# UNSUPERVISED LEARNING MODELS OF INVARIANT FEATURES IN IMAGES: RECENT DEVELOPMENTS IN MULTISTAGE ARCHITECTURE APPROACH FOR OBJECT DETECTION

Sonia Mittal

CSE Dept. Institute of Technology Nirma University, Ahmedabad

## *ABSTRACT*

*Object detection and recognition are important problems in computer vision and pattern recognition domain. Human beings are able to detect and classify objects effortlessly but replication of this ability on computer based systems has proved to be a non-trivial task. In particular, despite significant research efforts focused on meta-heuristic object detection and recognition, robust and reliable object recognition systems in real time remain elusive. Here we present a survey of one particular approach that has proved very promising for invariant feature recognition and which is a key initial stage of multi-stage network architecture methods for the high level task of object recognition.*

## *KEYWORDS*

*Unsupervised feature learning, CNNs, Tiled CNNs, Deep learning*

## 1. INTRODUCTION

A lot of research is being done in the area of object recognition and detection during last two decades. It involves multi-disciplinary field knowledge like image processing, machine learning, statistics /probability, optimization etc. In object detection and recognition the learning invariant representations of features is an important problem and it is the sub domain of computer vision.

The ability to learn robust invariant representations from a limited amount of labelled data is a critical step for the solution of the object recognition problem. In computer vision and pattern recognition domain various unsupervised methods are used to build feature extractor since long e.g. for dimensionality reduction or clustering, such as Principal Component Analysis and K-Means, have been used routinely in numerous vision applications [3].

The general framework for unsupervised feature learning (multistage)[4] is as below: It is divided broadly into two stages: Feature map learning and classification. First stage is the extraction of random patches from unlabelled training images. And a pre-processing would be done to the patches to normalize the data. After that learning a feature – mapping using an unsupervised learning. In the second stage consists of feature extraction, Pooling and classification. Feature would be extracted from equally spaced –sub patches covering the input

DOI :10.5121/ijscai.2016.5107

images, then pooling of features together over regions of the input images to reduce the number of feature values. And finally train a classifier to predict the labels given the feature vectors.

Primarily Multistage network for object detection/classification (high level computer vision tasks) are the Artificial Neural network, because of the shallow architecture (one input, hidden and out-put layer) [4, 5]. ANNs had not gain much of the success as it could be. Much of the momentum is not gain by this architecture due to the complexity and computation time required to execute back propagation methods for learning on relatively slow processor as compared to current and past years progress in speed of processor. But as the advent of the fast computational resources these types of architecture again become useful and with new variant i.e. Deep Learning Network came into existence. Deep learning Network were found very successful in computer vision problems.

Machine learning is the area where the method are used to provide the ability to machine which functions like humans Deep learning is one of the area of Machine learning methods i.e. it is based on learning representation of data. An image, video or audio can be represented as a vector of intensity value per pixel, or as a more abstract way as a set of edges, region of particular shape etc.as in Fig. 1. So by raw input, machine cannot understand about the nature of the data or characteristics of the data. If input data can be represented in some higher level like edges or shapes then machine can able to predict by combing these abstraction together [1,4,5]. Deep learning provides the way to represent input data into some higher level of abstraction as shown in Fig. 1. In deep learning methods main contribution was in 2006 by a few research groups, starting with Geoff Hinton's group, who initially focussed on starting unsupervised representation learning algorithms to obtain deeper representation [3, 5, 6, 11, 12, and 14].

Primarily our focus is on Convolutional Neural Networks (CNNs) and Deep Belief Networks (DBNs) because they are well established in the deep learning field and show great success. In literature many more variants of the convolutional networks are found, which are vary in how learning is done: supervised manner or unsupervised manner. Also [10] efforts was to develop invariant feature learning while extracting feature in same stage to reduce computation time. We will be discussing in detailed Tiled CNNs variant of CNNs in following section.

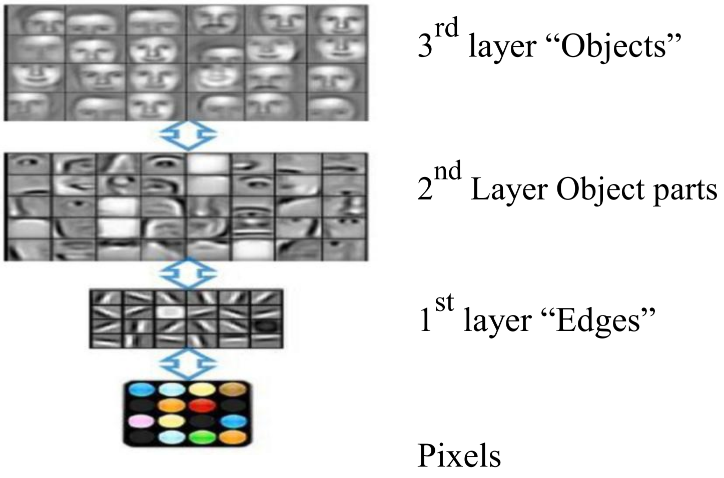

Fig.: 1 Feature Representation[22]

## 2. VARIOUS FRAMEWORKS

### 2.1 Deep 2learning: Deep belief networks (DBN)

Deep Belief network belongs to the class of generative model in machine learning. It can also be viewed as an alternating type of deep neural network, which is composed of multiple layer of latent/hidden variables [4]. Internally hidden layers are connected in a hierarchical manner to each other. But the units which are lying on the same layer are not connected to each other as shown in Fig.2. A DBN can learn statics of input data by providing the set of examples in an unsupervised manner. In this manner this kind of network can learn to probabilistically reconstruct its inputs. The hidden layers then act as feature detectors on inputs. After this learning step, a DBN can be further trained in a supervised way to perform classification.

In this type of networks depth of architecture indicates the number of levels of composition of non-linear operations in the function learned. As the researchers have gain the knowledge and insight into how our neuroscience functions [4], they are able to get the intuition about how the information is processed and communication is done in our brain and visual cortex, and are inspired by this model.

In [15] paper they have discussed about the stackable unsupervised learning system termed as encoder-decoder paradigm. Encoder will take raw data as an input and transform into representation known as code or feature vector (low dimensional). And after that a decoder will reconstructs the input from the representation as shown in Fig.4. Principle component analysis (PCA) Restricted Boltzmann Machine (RBM), Auto – encoder neural nets, Sparse energy based are the example of unsupervised learning methods which can be used to pre trained the Deep network. In [16] paper convolutional Auto-Encoders are discussed with the advantage that they can scale well to high dimensional inputs over auto-encoders which are used in DBN.

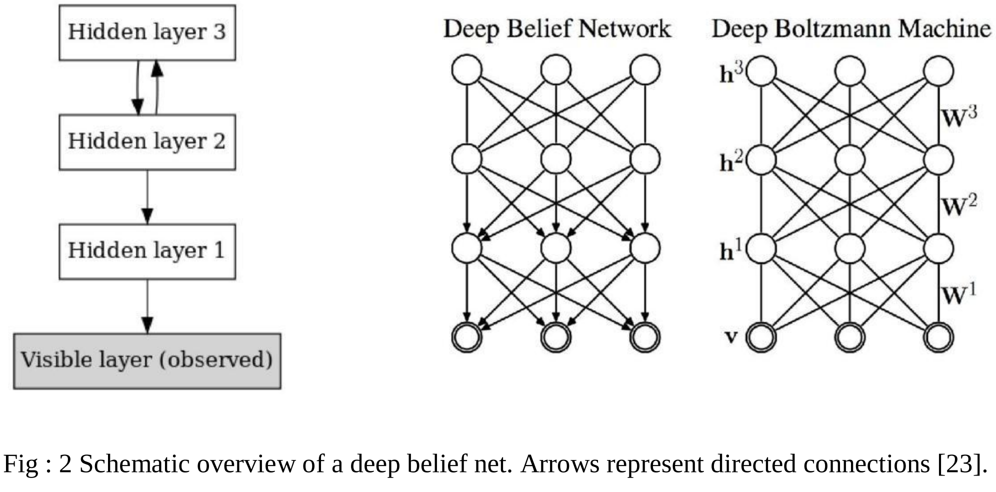

Fig : 2 Schematic overview of a deep belief net. Arrows represent directed connections [23].

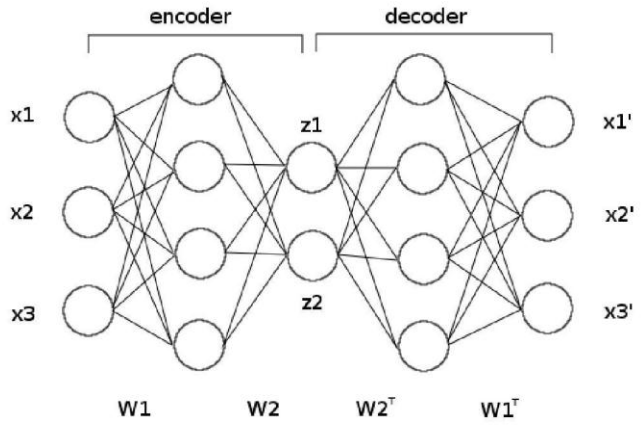

Fig : 3 Auto- encoder [24]

While fully connected Deep architecture do not scale well to realistic-sized high dimensional images in context to the performance and computational time. In Convolutional neural network discussed in next section their number of parameters which are used to describe their shared weights does not depends on the input dimensionality [19, 20, 21].

In paper [11] Hinton et. al have introduced a fast unsupervised learning algorithm for deep belief network based on greedy layer wise approach. It is a generative model as describe earlier with many hidden variables in hidden units. In DBN's RBM (Restricted Boltzmann machine) are generally used as basic building blocks, which is based on energy models. RBMs are usually trained using the contrastive divergence learning procedure (Hinton, 2002).

## 2.2 Convolutional Network

The fundamental concept behind the Convolution Network is biologically inspired trainable architecture. It is also the type of deep network having the ability to learn invariant features. Internal representations are hierarchical. In vision, pixels are assembled into edges, edges into motifs, motifs into parts, parts into objects, and objects into scenes as shown in Fig. 1.

This gives the motivation that recognition architectures for vision should have multiple trainable stages stacked on top of each other, one for each level in the feature hierarchy. Each stage in a ConvNets is consists of a filter bank, some non-linearities, and feature pooling layers. With multiple stages, a ConvNet can learn multi-level hierarchies of features as shown in Fig. 4.

While ConvNets have been successfully deployed in many commercial applications from OCR to video surveillance. Initially the convolution network was trained by Gradient-based supervised learning. The problem with Convolutional network is that it requires a large number of examples per class for training. If lower layers would be trained in an unsupervised manner then number of training samples would be reduced considerably. In last few years several research works have shown the advantages (in terms of speed and accuracy) of pre-training each layer of a deep network in unsupervised mode, before tuning the whole system with a gradient-based algorithm [3,9].

In many of the methods they incorporates invariance at its core, e.g. local contrast normalization (LCN) method, which introduces invariance with respect to illumination changes. That means method for unsupervised training of invariant feature hierarchies are designed. Once high-level invariant features have been trained with unlabeled data, a classifier can use these features to classify images through supervised training on a small number of samples.

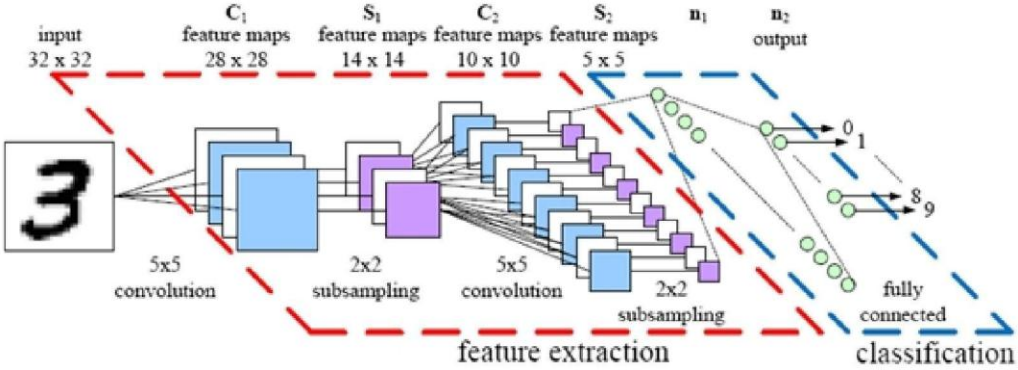

Fig. 4 : A typical Convolution architecture with two stages[25]

In Convolutional network or Convolutional neural network (CNNs) translated versions of the same basis function is used, and translation invariant features are built by pooling them. In CNNs the same basis function is shared across different image locations i.e. termed as weight tying CNNs. As a result of this technique have significantly fewer learnable parameters which makes it possible to train them with fewer examples than if entirely different basis functions were learned at different locations (untied weights).

Furthermore, Translation invariance is hard coded in CNNs, it is both advantageous and non-advantageous. Drawback of this hard-coding approach is that the pooling architecture captures only translational invariance; the network does not, for example, pool across units that are rotations of each other or capture more complex invariances, such as out-of-plane rotations. So it is better to let the network learn its own invariances from unlabelled data in place of hard-code translational invariance in the network. Tiled Convolutional network is the architecture which employs this scheme. Tiled convolutional neural network is an extension of Convolutional neural network that support both unsupervised pertaining and weight tiling.

## 2.3 Tiled Convolutional Network

The tiled convolutional networks (Tiled CNNs) [7], which uses a tiling scheme in which weight would be tied across the k steps away from each other. By using tied weight mechanism it has the benefit of significantly reducing the number of learnable parameters while giving the algorithm flexibility to learn other invariances. This method is based on only constraining weights/basis functions k steps away from each other as shown in fig : 5 CNN can be considered as a special case of Tiled CNN, with K=1 then it corresponds to Convolution network.

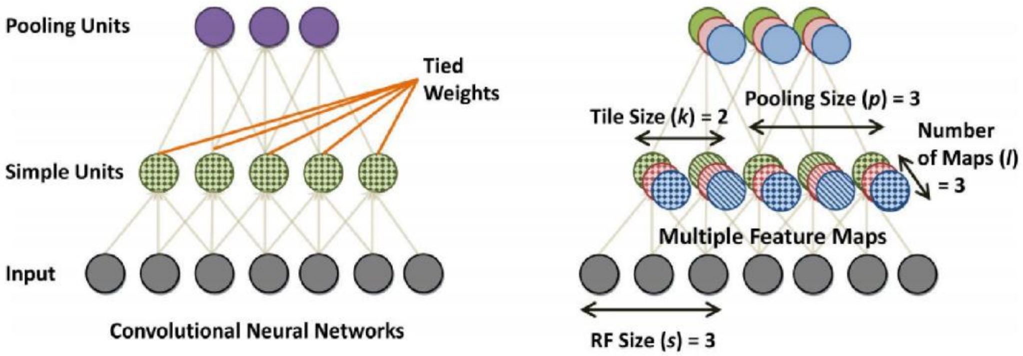

Fig. 5: Left: CNN Right: Partially untied local receptive field.

Figure 5 [26] can be understand as: left figure is showing the connectivity in the CNN; where local receptive fields and weights are tied. Right figure shows that in Tiled CNNs units same colour belong to same map; within each map, units with the same fill texture have tied together. ( Right fig: The circles which are on front level of same color (blue), the middle ones are of same color(red) and last level are of same color(green)) .

Tiled CNNs [7] are based on two main concepts: first one is local receptive fields (RF), and another is weight-tying. The receptive field concepts gives the locality awareness, hence each unit in the network only "looks" at a small, localized region of the input image. Now second concepts of weight-tying additionally enforces that each first-layer (simple) unit shares the same weights (see Figure 4-Left). This will provide the reduction in the number of learnable parameters, and by pooling over neighbouring units further hard-codes translational invariance into the model.

By Tiling connection of receptive fields instead of full receptive filed as in CNN, computational complexity would be reduced reasonably and allows the network to grow with more input or classes.

Weight-tying mechanism is allowing hard coded translational invariance and hence more invariant features cannot be possible because pooling units is capturing only simple invariance (illumination), in place of complex invariance such as scale and rotation invariance.

In Tiled CNNs [7], rather than tying all of the weights in the network together, instead a method is used that leaves nearby bases untied, but far-apart bases tied. This lets second-layer units pool over simple units that have different basis functions, and hence learn a more complex range of invariances. The Tiled Convolutional network achieves competitive results on the NORB and CIFAR-10 object recognition datasets.

## 3. CONCLUSIONS

The convolutional neural networks inherently will not provide unsupervised pre training, whereas Deep Belief networks, Tiled CNN provide unsupervised pre training. The variant of deep belief network: Staked Auto encoders are helpful in pertaining deep network. The CNNs are discriminative methods whereas Deep belief Networks is generative method. Stacked Auto encoders are discriminative models but denoising encoders maps to generative models. Depending upon the requirements of the application the choice of the framework can be selected.

## REFERENCES

[1] Yann LeCun, Koray Kavukcuoglu and Cl´ement Farabet, “Convolutional Networks and Applications in Vision” in IEEE, 2010

[2] K. Fukushima and S. Miyake, “Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position,” Pattern Recognition, vol. 15, no. 6, pp. 455–469, 1982.

[3] Ranzato, Fu Jie Huang, Y-Lan Boureau, Yann LeCun, “Unsupervised Learning of Invariant Feature Hierarchies with Applications to Object Recognition” in IEEE, 2007

[4] Itamar Arel, Derek C. Rose, Thomas P. Karnowski, “Deep Machine Learning—A New Frontier in Artificial Intelligence Research”, IEEE Computational Intelligence Magazine 2010

[5] H. Lee, R. Grosse, R. Ranganath, and A.Y. Ng, “Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations,” in ICML, 2009.

[6] A. Coates, H. Lee, and A. Y. Ng, “An analysis of single-layer networks in unsupervised feature learning,” in AISTATS 14, 2011.

[7] Q. V. Le, J. Ngiam, Z. Chen, D. Chia, P.W. Koh, and A. Y. Ng, “Tiled convolutional neural networks,” in NIPS, 2010.

[8] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Backpropagation applied to handwritten zip code recognition,” Neural Computation, 1989

[9] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Handwritten digit recognition with a back-propagation network,” in NIPS’89.

[10] K. Kavukcuoglu, M.A. Ranzato, R. Fergus, and Y. LeCun. Learning invariant features through topographic filter maps. In CVPR, 2009.

[11] G. E. Hinton, S. Osindero, and Y. W. Teh, “A fast learning algorithm for deep belief nets,” Neural Computation, 2006.

[12] G. E. Hinton and R.R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” Science, 2006.

[13] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, “Greedy layer wise training of deep networks,” in NIPS, 2007.

[14] Y. Bengio “Learning Deep Architectures for AI” Foundations and Trends in Machine Learning Vol. 2, No. 1 (2009) 1–127

[15] Marc'Aurelio Ranzato, Y-Lan Boureau and Yann LeCun: “Sparse feature learning for deep belief networks”, Advances in Neural Information Processing Systems (NIPS 2007), 2007

[16] J Masci, U Meier, D Cireşan, J Schmidhuber “Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction”, ICANN – 2011 Springer

[17] H. Lee, A. Battle, R. Raina, and Andrew Y. Ng, “Efficient sparse coding algorithms,” in NIPS, 2007.

[18] A. Krizhevsky, “Learning multiple layers of features from tiny images,” Tech. Rep., University of Toronto, 2009.

[19] Lee, H., Grosse, R., Ranganath, R., Ng, A.Y.: Convolutional deep belief networks for scalable nsupervised learning of hierarchical representations. In: Proceedings of the 26th International Conference on Machine Learning, pp. 609–616 (2009)

[20] Norouzi, M., Ranjbar, M., Mori, G.: Stacks of convolutional Restricted Boltzmann Machines for shift-invariant feature learning. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2735– 2742 (June 2009),

[21] Zeiler, M.D., Krishnan, D., Taylor, G.W., Fergus, R.: “Deconvolutional Networks” In: Proc. Computer Vision and Pattern Recognition Conference, CVPR 2010

[22] https://deeplearningworkshopnips2010.files.wordpress.com/2010/09/nips10-workshop-tutorial-final.pdf

[23] https://agollp.wordpress.com/2013/12/29/deep-belief-network/

[24] http://inspirehep.net/record/1252540/files/autoencoder.png

[25] http://parse.ele.tue.nl/cluster/2/CNNArchitecture.jpg

[26] http://amitpatelp.blogspot.in/2013/09/basic-convolutional-neural-network.html

## AUTHOR

Sonia Mittal is working as a Faculty with CSE Department for last 16 years. She has pursued her MCA from MBM Engineering College, Jodhpur. Her area of interest are Parallel Computing and Multimedia Processing. She has published 2 Research papers in International Journal. She has also involved in guiding major projects at post graduate level. Currently she is working as an Assistant Professor in Computer Science and Engineering department.