ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 7, Issue 1 , January 2020

Deep Learning Models, Architectures and Libraries for Implementing Pedestrian Detection

K. R. Sri Preethaa, A. Sabari

Assistant Professor / Department of CSE, KPR Institute of Engineering and Technology, Coimbatore, India.
Professor / Department of IT, K S Rangasamy College of Technology, Tiruchengode, India

ABSTRACT:In recent years, research in the pedestrian detection has gain its attention as a practical application of object detection. There exists a number of traditional pedestrian detection algorithms which requires expert design features to describe the characteristics of pedestrian and classifiers to classify them. Traditional filter based techniques on images like Histogram of Oriented Gradients filters and Machine Learning algorithms volume of input images, but suffers to perform well for large volume of data. With advancements in deep learning algorithms, huge volume of image data can be handled exponentially well. Convolutional Neural Networks (CNN) have made remarkable success in handling image, video and audio data. CNN has become important component for deep learning producing great success on mages and audio. Artificial designed methods of feature extraction suffer deficient description of pedestrian with complex background. This paper gives a consolidated study on availability of Deep learning models, its architectures and libraries for employing pedestrian detection.

KEY WORDS: Machine Learning, Deep Learning Convolutional Neural Networks, Recurrent Neural Networks.

I.INTRODUCTION

Pedestrian detection is the process of identification of presence of human in an image or video. It is also recommended in many applications to track the specific location of people travelling. Many of the smart cities require applications on locating pedestrians on the streets. This detection is considered as primary task in many applications like autonomous cars using cameras, smart city forensic surveillance, and hospital surveillance for patients and so on. Though many researches are carried out in pedestrian detection, it is still remains a challenging task, as real world applications require very high detection rate under complex environment.

With the recent developments in computer vision, the pedestrian detection in the intelligent monitoring, intelligent auxiliary driving, pedestrian analysis and intelligent robots are widely used. Though there where improvements in accuracy, pedestrian detection in critical conditions like presence of overlapped objects, presence of humanoid shaped objects, detecting humans at greater distance needs meticulous designing and optimization

Pedestrian detection methods has evolved over decades with variety of measures [1-4]. Some set of the methods focused increasing the accuracy of detections [5,6] and another set focused on increasing the detection speed [1,3]. The rapid development of computer hardware and software enabled the fame of deep learning technology. Especially, Convolutional Neural Networks (CNN) have evolved as the state-of-the-art technology in the obtaining accuracy in computer vision related tasks. The methods based on deep learning has proved increased performance with wide margin.

The paper mainly focuses on importance, applications and techniques used for pedestrian detection. Section II focuses survey on approaches for pedestrian detection. Section III gives an overview of Deep Learning Models available for pedestrian detection. Section IV explains the methodology of pedestrian detection using CNN. Section V section discusses libraries available for pedestrian detection, followed by applications of pedestrian detection and conclusion.

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 7, Issue 1 , January 2020

II. PEDESTRIAN DETECTION APPROACHES

A number of approaches has been proposed for pedestrian detection in the literature. The following discusses some of the most commonly used approaches.

A. Holistic detection

In this approach the detectors are trained to detect pedestrians in each image frame of sequence of video frames. The detector would indicate in search window, if the image features meet the criteria of a pedestrian. Some methods employs features globally like in edge template [1], while others use local features like histogram of gradients descriptors [2]. However these approaches suffer from performance degradation due to occlusions and background clutter in images.

B. Part-based detection

In part based detection method, pedestrians are modelled as collections of parts. Each part hypothesis is generated by learning its local features that includes edgelet [3] and Orientation features [4].These parts hypotheses are the combined together to form pedestrian hypothesis. Though this method is simple, attaining accuracy in part detection is difficult. The implementation process requires a standard procedure of creating a sample pyramid of images, extracting features at each scale, classifying at all locations and generating a set of bounding box [5].

C. Patch-based detection

Leibe et al. [6] has proposed Implicit Shape Model (ISM) by combining detection and segmentation approaches. During training process the local appearance is learnt as a codebook. The extracted local features are matched against the codebook during the detection process. Here each match is given a vote for pedestrian detection. Further refining on these hypotheses to produce a final detection result. This method requires only smaller number of training image data.

D. Motion-based detection

Motion based detection model employs subtraction of back ground (where no motion) or foreground (with motion) pixels of video streams. This procedure spots the silhouettes of every moving element in a scene along with the humans. At university of Liege, an algorithm [7] [8] is developed to detect humans by analyzing the shapes of silhouettes. As these methods considers silhouette as a whole and makes a single classification, it is sensitive to shape defects. It is proposed to split silhouetteinto parts as set of smaller region to decrease the defect detections. On other hand in some part based approaches the small set of regions do not have anatomical meaning. This paved way to extended algorithms for pedestrian detection in 3D video streams [9].

E. Detection using multiple cameras

Fleuret et al. [10] has formulated a method by integrating multiple calibrated cameras for multiple pedestrian detection. The ground plane is divided into uniform, non-overlapping grid cells with 25 X 25 (cm) in dimension. A Probability Occupancy Map (POM) is produced by the detector that probability of presence of a person in a grid. The videos streams are taken at different angles at eye level from multiple cameras. This method effectively combines a generative model with dynamic programming to follow up across thousands of frames produces metrically accurate trajectory in spite of significant occlusion and lightening changes.

III. DEEP LEARNING MODELS FOR PEDESTRIAN DETECTION

The accuracy and efficiency in performance of pedestrian detection with the machine learning algorithms are bounded small sized data. Real time applications require handling of a huge amount of data and increased performance. This can be achieved through deep learning models.

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 7, Issue 1 , January 2020

A. Multi-Layer Perceptron

The Multilayer Perceptron's (MLPs) has at least of three layer of operations [19]. Each layer consists of densely connected units and each require substantial amount of weights attached to it. MLP is the initial stage of Artificial Neural Network (ANN). MLP containing more number of hidden layer s is called deep learning structures. The operation at standard MLP with a vector x is shown in below equation 1.

$$\mathbf{y} = \sigma(W \cdot \mathbf{x} + b) \quad \ldots\ldots \quad 1$$

Where x in an input vector, y is the output of the layer, w is the weight of each unit, b is a biasing factor and σ denotes activation function that aims to improve the non- linearity of the model.

B. Boltzmann Machine

Restricted Boltzmann Machines (RBMs) [20] are an energy based graphical models, originally designed for unsupervised learning purpose. It includes a hidden layer, a visible layer and each unit can have only binary values (0 and 1).

The probabilities of these values are given in equation 2:

$$P(h_j = 1|v) = \frac{1}{1 + e^{-\mathbf{W}\cdot\mathbf{v}+\mathbf{b_j}}}$$
$$P(v_j = 1|h) = \frac{1}{1 + e^{-\mathbf{W^T}\cdot\mathbf{h}+\mathbf{a_j}}}. \quad \ldots\ldots\ldots \quad 2$$

Where *h*, v denotes hidden and visible units respectively, w are the weights of each unit, a and **b** are biases. The visible and hidden units are conditional independent of each other.

C. Auto-Encoders

Auto Encoders (AEs) replicates inputs to outputs and are designed for unsupervised learning. AEs are used to learn compressed representation of data for reducing dimensions [21], Extended versions like Denoising Auto-Encoder (DAE) [22], Variational Auto-Encoders (VAEs) [23] are engaged for initializing weights in deep learning models and for generation of virtual examples from target data distribution respectively.

D. Convolutional Neural Network

Convolutional Neural Networks (CNNs) are multi layered structures that employs a group of locally connected filters of layers to obtain the correlation between the different data regions. The following equation 3 represents the mathematical working model of standard convolution for each location p_y of outputy

$$\mathbf{y}(\boldsymbol{p}_y) = \sum_{\boldsymbol{p}_G \in \mathbb{G}} \mathbf{w}(\boldsymbol{p}_G) \cdot \mathbf{x}(\boldsymbol{p}_y + \boldsymbol{p}_G) \quad \ldots.. \quad 3$$

Here P_G represents all positions in the receptive field G of the convolutional filter W. It also denotes the receptive range of each neuron to the inputs in a convolutional layer. Here the weights are shared across locations in input map

E. Recurrent Neural Network

Recurrent Neural Networks (RNNs) modeled such that there exist a sequential correlation between the samples. It is mainly designed for sequential data, where the recurrent connections between the hidden units produce output at each

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 7, Issue 1 , January 2020

step [24]. The mathematical representation of working of standard RNNs model, when a sequence of inputs x = {$x1$; $x2$; · · ·; xn} are given in equation 4.

$$s_t = \sigma_s(W_x x_t + W_s s_{t-1} + b_s)$$
$$h_t = \sigma_h(W_h s_t + b_h), \quad \ldots\ldots \quad 4$$

Here s_t stands for state of network at a given time t. It constructs memory unit for the network. The value of s_t is calculated using function on input xt and previous state S_{t-1} .The h_t denotes the network output at time t.

F. Generative Adversarial Network

The Generative Adversarial Network (GAN) is a frame work which trains two models simultaneously, a generative G that approximates the target data distribution from training data and a discriminative model D, which is determines the probability of the sample that it comes from the real training data than from the output of G [25]. So this framework is considered to be a generative model of adversarial process. The G and D models are neural networks, where the training process of G tries to increase the probability of D making mistakes. The GAN aims to solve the following minmax problem using equation 5. [25]

$$\min_{\mathcal{G}} \max_{\mathcal{D}} \mathbb{E}_{x\sim P_r(x)}[\log \mathcal{D}(x)] + \mathbb{E}_{z\sim P_n(z)}[\log(1 - \mathcal{D}(\mathcal{G}(z)))] \quad \ldots\ldots. \quad 5$$

IV. CNN FOR PEDESTRIAN DETECTION

Basically the image inputs to a CNN layer are RGB images of multiple 2D matrices of different regions. A convolutional layer uses a number of filters in different locations to scan inputs and to produce outputs. If there are M and N are the inputs and outputs filters respectively, then for convolution operation the convolutional layer requires 2 X 3 filters. Figure 1 below illustrates the operation of a 2D convolutional layer.

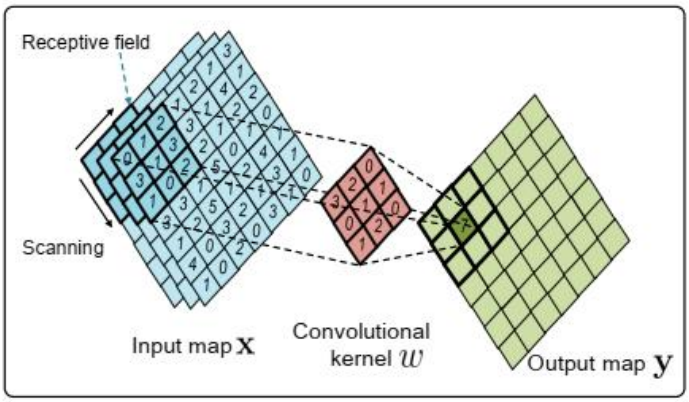

Figure 1: Operation Principle of a Convolutional Layer

A. PRINCIPLE OF CNN

CNNs expands over traditional MLPs on three mainly focusing on (a) spare interactions (b) Parameter sharing and (c) equivariant representations [25]. It significantly reduces the number of model parameters and preserves affine variance. The spare interactions shows that the weight kernel is smaller than the input size. The outputs are produced by moving the filters over the current layer. The same kernel is used to scan the whole map indicating parameter sharing. As the number of parameters is greatly reduced, the problem of over fitting gets diminished. Equivariant representations is very useful in image processing where the essential features are indicated in different locations of an image, with various affine patterns.

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 7, Issue 1 , January 2020

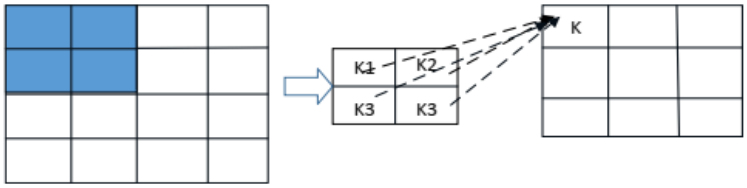

Figure 2. Convolution operation

CNNs has remarkable applications in image processing. . Krizhevsky *et al.* [26] has classified images on ImageNet dataset using a CNN [27]. He has reduced top 5 errors by 39.7% using the above method which paved way for revolution in field of image classification. Later GoogLeNet [28] and ResNet [29] has introduced inception and residual learning techniques that diminishes the over fitting problems and gradients by increasing the depth of CNN structures. The Dense Convolutional Network (DenseNet) [30] is a further enhanced structure that reuses feature map from each layer, achieving greater accuracy over other CNN models with fewer layers. CNNs are used for video applications involving object detection. Ji et al. has demonstrated that 3D convolutional neural network is superior to 2D CNN for a video activity recognition of pedestrians. Recent researches focuses on learning of convolutional kernels, which re dynamic architectures concentrates on important regions of input maps.

V. LIBRARIES FOR DEEP LEARNING

Recent years with growing popularity in AI, many real time applications employs deep learning models. Building deep learning models from scratch is complex as it requires definitions for behaviors and gradients at each layer. Many libraries has been dedicated to facilitate this process. Toolboxes working with multiple programming languages, along with GPU acceleration and parallelization are embedded for support. This section summarizes the some of the libraries supports for deep learning modeling.

A. TensorFlow

Google has developed a machine learning library namely TensorFlow for implementations on both single and distributed architectures. It facilitates deploying computation graphs on CPUs, GPUs and also on mobile devices [35, 36]. TensorFlow is also used for data driven research purposes. TensorFlow provides graphical visualization tools that helps users with easy understanding of models, data flow and helps in debugging. It supports multiple programming languages like python, java, C and Go along with high-level programming interfaces including Keras6, TensorLayer and Luminoth7 [37]. It enables fast implementations of neural networks on cloud and fog services. It is most popular deep learning library of current era.

B. **Theano**

Theano is a Python library that permits multidimensional data to define efficiently, optimize and evaluate numerical computations [38]. It enables users to deploy their programs on both CPU and GPU modes. Though understanding of Theano is quite difficulty, it helps in building efficient Neural networks with substantial compilation time.

C. Caffe

Caffe is developed by Berkeley AI Research [39], a devoted deep learning framework. The latest version is Caffe2.8, a flexible framework that enables building of models efficiently. It inherits all advantages of its old versions, allows to train neural networks on multiple GPUs with distributed environment and also supports machine learning on mobile operating systems. It has a great future endeavors in mobile edge computing.

D. (Py)Torch

PyTorch is a python version of scientific computing frame work that supports machine learning models and algorithms [40]. It is a light weighted toolbox that can be embedded on mobile platforms. Building Neural networks using PyTorch

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 7, Issue 1 , January 2020

is easier, provided with rich pre-trained models and modules that enables reusability and extendibility, hence it has gained popularity rapidly. It is mainly employed for research purposes and maintained by Facebook.

E. MXNET

MXNET is a scalable and flexible machine learning library that supports multiple programming languages. It supports both single and distributed environments with fast numerical computations. A standard neural network can be easily constructed into deep learning model without substantial coding effort. Its difficulty in learning has relatively reduced the number of users. It is official deep learning framework for Amazon. The following is the table summarizing the features of deep learning libraries discussed.

Library	Low-Layer Language	Available Interface	Pros	Cons	Mobile Supported	Popularity	Upper-Level Library
Tensor-Flow	C++	Python, Java, C, C++, Go	• Large user community • Well-written document • Complete functionality • Provides visualization tool (TensorBoard) • Multiple interfaces support • Allows distributed training and model serving	• Difficult to debug • The package is heavy • Higher entry barrier for beginners	Yes	High	Keras, TensorLayer, Luminoth
Theano	Python	Python	• Flexible • Good running speed	• Difficult to learn • Long compilation time • No longer maintained	No	Low	Keras, Blocks, Lasagne
Caffe(2)	C++	Python, Matlab	• Fast runtime • Multiple platform s support	• Small user base • Modest documentation	Yes	Medium	None
(Py)Torch	Lua, C++	Lua, Python, C, C++	• Easy to build models • Flexible • Well documented • Easy to debug • Rich pretrained models available • Declarative data parallelism	• Has limited resource • Lacks model serving • Lacks visualization tools	Yes	High	None
MXNET	C++	C++, Python, Matlab, R	• Lightweight • Memory-efficient • Fast training • Simple model serving • Highly scalable	• Small user base • Difficult to learn	Yes	Low	Gluon

Table 1: Features of Deep learning Libraries

VI. METHODOLOGY AND APPLICATIONS

Pedestrian detection is a process of recognizing an object in a given image or video as human or not. It requires a number of preprocessing steps which includes frame extractions and dimensional compression. Figure 1 given below illustrates the methodology of Pedestrian detection process. The model gets input as video from a real time surveillance camera or an already recorded video. The input video is converted into sequential image frames. Features of human are extracted for the image using a number of techniques like HOGs to build an object recognition classifier model. Each extracted features points are used to identify parts of the human like head, arms, limbs etc.. The combined identified parts are used for recognition as a human. This process requires pre trained model and learning phase. The learning task implemented on Convolutional Neural Networks implemented using deep learning libraries. The testing and validation phase is carried out enhance the accuracy of the model.

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 7, Issue 1 , January 2020

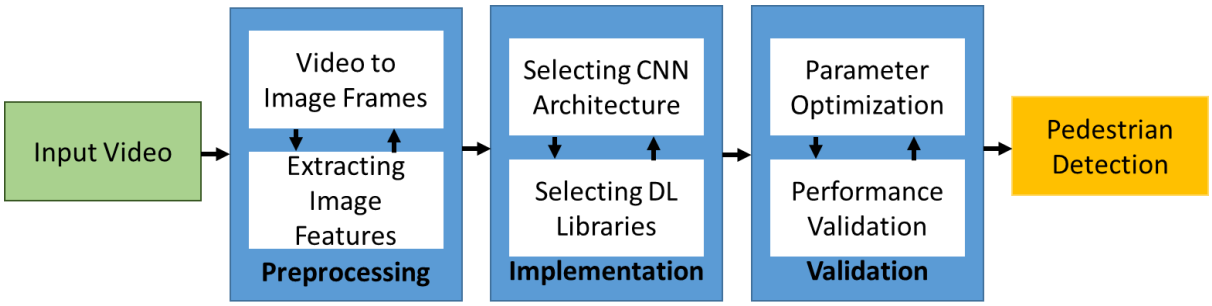

Figure 1: Pedestrian Detection Methodology

APPLICATIONS

A. Pedestrian Detection in Autonomous driving Vehicles

With the growing popularity of object detection, many real time applications has been developed. Pedestrian detection is an act of identifying presence of a human in an image or video. This is embedded in autonomous vehicular systems where the driverless vehicles act according to their environment based on the presence of pedestrians in their bounded environment. The image or video captured is given to the learning architecture to detect the presence of pedestrian as well it is necessary to detect their activity. It is surveyed that more than 5,000 pedestrains lost their lives in road accidents. It is also suggested to embed activity and behavioral analysis of people so as to avoid such road accidents.

B. Automatic capturing of web lectures and presentations

Recording a lecture or presentation in a camera requires human intervention. Automatic capturing of web lectures enables capturing of web lectures automatically without camera man, following the presenter. Though this system has a number of challenges like occlusion which requires the use of multiple models (face, full body, upper body,) to be combined, it has a plenty of applications in real time.

C. Detection of pedestrians around AGVs and in the blind spot area of trucks

It is a more challenging task for detecting pedestrians around Autonomous Vehicles and in the blind spot area of trucks. It is very important to have a greater accuracy in detection to avoid false detections in detecting all vulnerable road users. False detections may lead to missing detection that may cause deadly accidents, false alarm, etc. Additionally is highlighted that detection has time constraints too. The main goal of pedestrian detection is to provide safety to both drivers and road users.

VII CONCLUSION

This paper discusses the various available deep learning models and architectures. The methodology for pedestrian detection using deep learning models is discussed. Also it provides various libraries available for implementing deep learning in python. This work will help the researchers further in implementing pedestrian detection in applications.

REFERENCES

[1] C. Papageorgiou and T. Poggio, 'A Trainable Pedestrian Detection system", International Journal of Computer Vision (IJCV), pages 1:15–33, 2000.
[2] N. Dalal, B. Triggs, 'Histograms of oriented gradients for human detection', IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pages 1:886–893, 2005.
[3] Bo Wu and Ram Nevatia, 'Detection of Multiple, Partially Occluded Humans in a Single Image by Bayesian Combination of Edgelet Part Detectors', IEEE International Conference on Computer Vision (ICCV), pages 1:90–97, 2005.
[4] Mikolajczyk, K. and Schmid, C. and Zisserman, A. 'Human detection based on a probabilistic assembly of robust part detectors', The European Conference on Computer Vision (ECCV), volume 3021/2004, pages 69–82, 2005.
[5] Hyunggi Cho, Paul E. Rybski, Aharon Bar-Hillel and Wende Zhang 'Real-time Pedestrian Detection with Deformable Part Models'.

ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 7, Issue 1 , January 2020

[6] B.Leibe, E. Seemann, and B. Schiele. 'Pedestrian detection in crowded scenes',IEEE Conference on Computer Vision and Pattern Recognition(CVPR), pages 1:878–885, 2005.
[7] O. Barnich, S. Jodogne, and M. Van Droogenbroeck. 'Robust analysis of silhouettes by morphological size distributions',Advanced Concepts for Intelligent Vision Systems(ACIVS), pages 734–745, 2006.
[8] S. Piérard, A. Lejeune, and M. Van Droogenbroeck. 'A probabilistic pixel-based approach to detect humans in video streams', IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 921–924, 2011.
[9] 9.S. Pierard, A. Lejeune, and M. Van Droogenbroeck. '3D information is valuable for the detection of humans in video streams', Proceedings of 3D Stereo MEDIA, pages 1–4, 2010.
[10] F. Fleuret, J. Berclaz, R. Lengagne and P. Fua, Multi-Camera People Tracking with a Probabilistic Occupancy Map, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 30, Nr. 2, pp. 267–282, February 2008.
[11] Martin Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. TensorFlow: A system for large-scale machine learning. In USENIX Symposium on Operating Systems Design and Implementation (OSDI), volume 16, pages 265–283, 2016.
[12] Moustafa Alzantot, Yingnan Wang, Zhengshuang Ren, and Mani B Srivastava. RSTensorFlow: GPU enabled tensorflow for deep learning on commodity Android devices. In Proc. 1st ACM International Workshop on Deep Learning for Mobile Systems and Applications, pages 7–12, 2017.
[13] Hao Dong, Akara Supratak, Luo Mai, Fangde Liu, Axel Oehmichen, Simiao Yu, and Yike Guo. TensorLayer: A versatile library for efficient deep learning development. In Proc. ACM on Multimedia Conference, MM '17, pages 1201–1204, 2017.
[14] Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. arXiv e-prints, abs/1605.02688, May 2016.
[15] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, SergioGuadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093, 2014.
[16] R. Collobert, K. Kavukcuoglu, and C. Farabet. Torch7: A Matlab-like environment for machine learning. In Proc. BigLearn, NIPS Workshop, 2011.
[17] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. 2017.
[18] Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. arXiv preprint arXiv:1512.1274, 2015.
[19] Ronan Collobert and Samy Bengio. Links between perceptrons, MLPs and SVMs. In Proc. twenty-first ACM international conference on Machine learning, page 23, 2004.
[20] Nicolas Le Roux and Yoshua Bengio. Representational power of restricted boltzmann machines and deep belief networks. Neural computation, 20(6):1631–1649, 2008.
[21] Yoshua Bengio et al. learning deep architectures for AI. Foundations and trends R in Machine Learning, 2(1):1–27, 2009.
[22] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of Machine Learning Research, 11(Dec):3371–3408,2010.
[23] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In Proc. International Conference on Learning Representations (ICLR), 2014.
[24] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep learning. MIT press, 2016.
[25] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Advances in neural information processing systems, pages 2672–2680, 2014.
[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012.
[27] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. International Journal of ComputerVision (IJCV), 115(3):211–252, 2015.
[28] 28. Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In Proc. IEEE conference on computer vision and pattern recognition, pages 1–9, 2015.
[29] 29. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proc. IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.
[30] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten, 'Densely connected convolutional networks', IEEE Conference on Computer Vision and Pattern Recognition, 2017.
[31] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3D convolutional neural networks for human action recognition. IEEE transactions on pattern analysis and machine intelligence, 35(1):221–231, 2013.
[32] Junmo Kim Yunho Jeon. Active convolution: Learning the shape of convolution for image classification. In Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2017.
[33] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In Proc. IEEE International Conference on Computer Vision, pages 764–773, 2017.
[34] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable ConvNets v2: More Deformable, Better Results. arXiv preprint arXiv:1811.11168, 2018.
[35] Wencong Xiao, Jilong Xue, Youshan Miao, Zhen Li, Cheng Chen, Ming Wu, Wei Li, and Lidong Zhou. Tux2: Distributed graph computation for machine learning. In USENIX Symposium on Networked Systems Design and Implementation (NSDI), pages 669–682, 2017.
[36] Paolini, Monica and Fili, Senza . Mastering Analytics: How to benefit from big data and network complexity: An Analyst Report. RCR Wireless News, 2017.

ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 7, Issue 1 , January 2020

[37] Chaoyun Zhang, Pan Zhou, Chenghua Li, and Lijun Liu. A convolutional neural network for leaves recognition using data augmentation. In Proc. IEEE International Conference on Pervasive Intelligence and Computing (PICOM), pages 2143–2150, 2015.
[38] Richard Socher, Yoshua Bengio, and Christopher D Manning. Deep learning for NLP (without magic). In Tutorial Abstracts of ACL 2012, pages 5–5. Association for Computational Linguistics.
[39] IEEE Network special issue: Exploring Deep Learning for Efficient and Reliable Mobile Sensing. http://www:comsoc:org/netmag/cfp/ exploring-deep-learning-efficient-and-reliable-mobile-sensing, 2017.
[40] Mowei Wang, Yong Cui, Xin Wang, Shihan Xiao, and Junchen Jiang. Machine learning for networking: Workflow, advances and opportunities. IEEE Network, 32(2):92–99, 2018.
[41] Mohammad Abu Alsheikh, Dusit Niyato, Shaowei Lin, Hwee-Pink Tan, and Zhu Han. Mobile big data analytics using deep learning and Apache Spark. IEEE network, 30(3):22–29, 2016.
[42] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep learning. MIT press, 2016.

AUTHOR'S BIOGRAPHY

K. R. Sri Preethaa Assistant Professor (Sr.G) in Department of Computer Science and Engineering, KPR Institute of Engineering and Technology, Coimbatore. She has 10 years of teaching experience. Her area of research includes Computer Vision, Data analytics, sentiment analysis, Information retrieval and SEO on which she has published over 21 technical papers in conferences and journals.

A. Sabari, professor and Head of Department of Information Technology, K.S. Rangasamy College of Technology, Tiruchengode, India. He has completed his Masters of Engineering in Information Technology, Masters of Business Applications. He completed his research in the Department of Information and Communication. He has twelve years of teaching experience and four years of research experience. He has published over 29 technical papers in various international journals and conferences. His areas of interest include mobile adhoc network, data mining and analytics.