



REVUE ALGERIENNE DES SCIENCES

Section -A-

Sciences de la Nature et de la Vie

Sciences techniques

<http://univ-eltarf.dz/fr/>



Handwritten Digit Recognition: Developing an Efficient ML and DL Model to Recognize Handwritten Digits

Tahmi Hassina, Said Gadri, Baya Loues, Adouane Nour El-Houda, Bouafia Nadjet

University Mohamed Boudiaf of M'sila, Algeria

Informations	Abstract
Keywords: machine learning, deep learning, handwritten digit recognition, Pattern Recognition, Neural Networks, Convolutional Neural Networks.	Deep learning DL is a new subfield of machine learning ML area which is used during the last decades to develop more sophisticated algorithms allowing high performance in some popular recognition fields, such as: pattern recognition, computer vision and image classification. Among the most used methods in DL, we find CNNs (Convolutional Neural Networks) which can be considered as the best used technique. In the present work, we have developed an automatic classifier that permits to classify some given grayscale images representing handwritten digits into one of 10 classes (digits from 0 to 9), inclusively. For this purpose, we have used ML and DL approaches. First, we proceeded to the classification task using many ML algorithms including: LR, LDA, KNN, CART, NB, and SVM. Second, we proposed a new CNN model composed of many convolutional layers. Finally, we established a comparison between different algorithms.
*Correspondence : Hassina.tahmi@univ-msila.dz	

1. Introduction

The main idea of ML algorithms is how to perform important tasks by generalizing from a large number of given examples. So, ML systems automatically learn programs from large amount of data. In the last years, many IA tasks such as: spam filtering, web search, text categorization, recommender systems, credit scoring, fraud detection, etc, are based essentially on Machine Learning approaches ML which gave a high performance. The McKinsey Global Institute asserts that ML discipline will be the guide of the next big wave of innovation [16], [17], [25]. Currently, we are able to use larger datasets to learn more powerful models, and better techniques to avoid overfitting and underfitting. Many algorithms have been developed in this area of research, including: logistic regression, k-nearest neighbors, naïve Bayes, decision trees, support vector machine, artificial neural networks, etc. Researches on Artificial Neural Networks ANN can be considered as the oldest discipline in ML and AI that dates back to McCarthy in 1943, but these researches were quickly interrupted for a long time due to their high requirements in term hardware, software and running time. Many years later, they were revived in parallel with the apparition of new sub-field called deep learning DL. The DL approach greatly simplifies the feature

engineering process especially in some vital areas such as: medical imaging analysis. Among the DL methods, CNNs are of special interest. When exploiting local connectivity patterns efficiently which is the case of those used in the ImageNET competition [8]. There are many works trying to apply CNNs on image analysis [9, 10] using a variety of methods like the rectified linear unit [11] and deep residual learning [12]. The organization of the rest of the paper is the following. In Section 2, we introduce the Deep Learning Background. In section 3 and 4, then Proposed Model, Experimentation, also a famous MNIST dataset using TensorFlow, keras and python as language and its libraries. Illustration of Obtained Results as user enters the respective digit the machine would recognize and show the results with accuracy percentage in Section 5. In section 6 and 7, presenting Discussion, Comparison between ML and CNN approaches. Finally, Section 8 presents some concluding remarks.

2. Deep Learning Background

During the last decade, Deep Learning DL becomes one of the most popular sub-field of IA and ML, especially in speech recognition, computer vision and some other interesting topics. Its success is motivated by three factors: the increased amount of available data, the

improvement and the lowest cost of hardware and software [9, 10, 11, 12, 13], the increased chip processing abilities (e.g., GPU units) [14]. DL is based essentially on the use of ANNs with two layers or many hidden layers. The convolutional neural network CNN is a specialized feedforward neural network that was developed to process multidimensional data, such as images. Its origins refer to the neocognition proposed by Fukushima in 1980 [15]. The first model of CNN was proposed by LeCUN et al., [16] in 1998 for the purpose of character recognition. Furthermore, many other alternative ANN architectures have been developed later, including: recurrent neural networks RNNs, autoencoders, and stochastic networks [17, 18, 19]. Deep learning also refers an efficient solution to the problem of input data representation which is a critical phase in ML especially for complex problems such as image and speech recognition [20]. DNNs are able to learn high level feature representations of inputs through their multiple hidden layers. The first DNNs had appeared in 1960's, but abandoned, after that for long time in favor of ML approach, due to its high requirements in term difficulties in training and inadequate performance [21]. In 1986 Rumelhart et al., [22] proposed the back-propagation method to update efficiently neural network weights using the gradient of the loss function through multiple layers. Despite the promising results given by DNNs in the late of 1980's [23] and 1990's [24], they were abandoned in practice and research due to many problems. In 2006, researches in DL were revived especially when some new methods for sensibility initializing DNN weights have been developed by researchers [25, 26]. It is the case of deep belief networks DBNs which proved their efficiency in image and speech tasks in 2009 and 2012 [11]. In 2012, Krizhevsky et al., [17] proposed a deep CNN for the large-scale visual recognition challenge (LSVRC) [27] reducing the error rate from the previous year's 26% down to just 16%. This CNN has been implemented on multiple graphics processing units GPUs for the first time, This new technique has allowed the training of large datasets and increase significantly the speed of processing and the research productivity. Furthermore, the use of a new activation function RELU (Rectified Linear Unit) has ended the problem of gradient and allowed faster training of data. Dropout technique is also used as a regularization method to decrease overfitting in large networks with many layers. All these interesting improvements and the increasing internet in DL let the leading technical companies to increase the research efforts, producing many other advances in the field. Many DL frameworks abstracting have

tensor computation [12, 13, 14, 15] and GPU compatibility libraries [16] have been developed and made available to re-researchers through open source software [28] and cloud services [29, 30]. On the other hand, many companies have met the challenges of big data when exploring large amounts of data to predict value decisions [30]. The concept of big data refers to data that exceeds the capability of standard data storage and data processing systems [31]. This large volume of data requires also high-performance hardware and very efficient analysis tools. Some other ML challenges appeared with big data including: high dimensionality, distributed infrastructures, real time requirements, feature engineering. Najafabadi et al [32] discuss the use of DL to solve big data challenges, the capacity of DNNs to extract meaningful features from large sets of unlabeled data is extremely important as it is commonly encountered in big data analytics. The automatic extraction of features from unstructured and heterogeneous data, e.g., image text, audio is very useful and difficult task. But this task becomes easy with the use of DL methods. Other tasks including: advanced semantic-based information retrieval systems like semantic indexing and hashing [33, 34] are also became possible with these high-level features, furthermore, DL is also used to tag incoming data streams allowing to classify and organize fast moving data [32]. In general, high capacity DNNs are suitable for learning from the large volumes of data issued of big data sources. As conclusion, we can say that DL is currently growing faster than ever before.

3. The Proposed Model

In the present work, we have developed an automatic classifier that permits to classify some given grayscale images representing handwritten digits into one of 10 classes representing integer values from 0 to 9, inclusively based on two approaches: the classic ML approach and DL approach. Below shown is a small workflow of how CNN module will extract the features and classify the image based on it. The architecture shows the input layer, hidden layers and output layer of the network. There are many layers involved in the feature extraction phase of the network which involves convolution and subsampling. Initially, we proceeded to the classification task using many ML algorithms including: LR, LDA, KNN, CART, NB, and SVM. Then we proposed a CNN model composed of many convolutional layers, many Max-Pooling layer and one full connected layer. Finally, we established a comparison between the different algorithms. As programming tools,

we have used Python, Tensor-flow, and Keras which are the most used in this field. Fig 1 summarize the classification task based on the two approaches, while Fig 2 presents a detailed diagram of the proposed CNN model in order to improve the performance of the classification task.

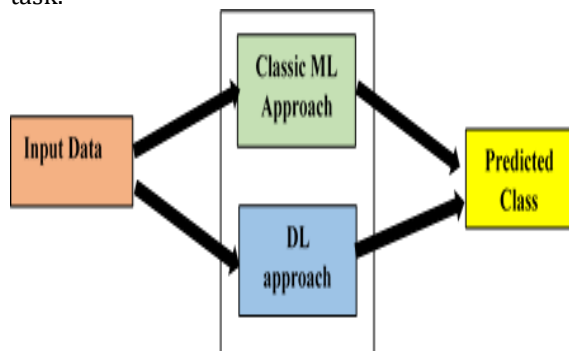


Fig 1. Handwritten Digit Classification Process (big size)

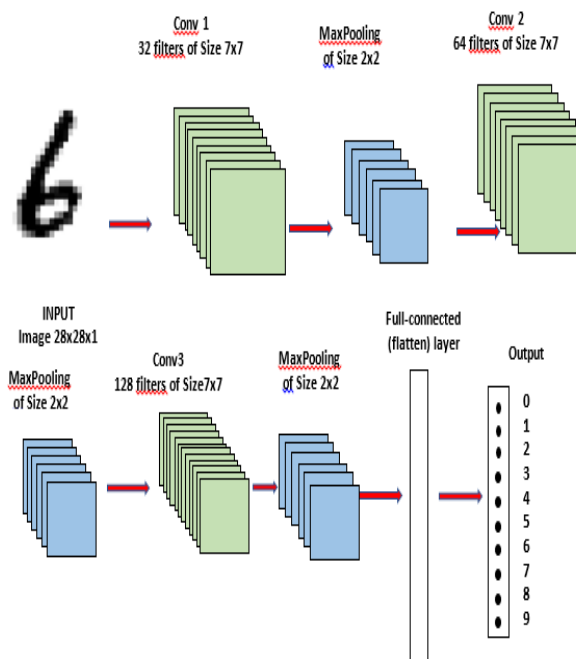


Fig 2. Architecture of the proposed CNN model

4. Experimentation

4.1. Used dataset

In our experiments, we have used Digits Mnist dataset contains 60,000 small square 28×28 pixels grayscale images of handwritten single digits between 0 and 9.

4.2. Programming Tools

Python: Python is currently one of the most popular languages for scientific applications. It

has a high-level interactive nature and a rich collection of scientific libraries which lets it a good choice for algorithmic development and exploratory data analysis.

Tensorflow: TensorFlow is a multipurpose open source library for numerical computation using data flow graphs.

Keras: Keras is the official high-level API of TensorFlow.

4.3. Evaluation

In order to validate the different ML algorithms, and obtain the best model, we have used the cross-validation method consisting in: splitting our dataset into 10 parts, train on 9 and test on 1 and repeat for all combinations of train/test splits. For CNN model, we have used two parameters which are: loss value and accuracy metric.

5. Illustration of Obtained Results

To build the best predictive model and achieve the higher accuracy rate, we have performed two tasks:

1. Applying many classic ML algorithms, including: Logistic Regression LR, Linear Discriminant Analysis LDA, K-nearest Neighbors KNN, Decision Tree (CART variant), Gaussian Naïve Bayes NB, Support Vector Machine SVM. For this purpose, we used scikit-learn library of python containing the most known learning algorithms.

2. Designing a CNN (Convolutional Neural Network) model composed of many layers as it is explained in Figure 1:

- ✓ A first convolutional layer Conv1 constituted of 32 filters of size (7x7).
- ✓ A dropout layer to avoid overfitting.
- ✓ A MaxPooling MaxPool of size (2x2) allowing to reduce dimensions (weigh, high) of images issued of the previous layer after applying the different filters of Conv1.
- ✓ A second convolutional layer Conv2 constituted of 64 filters of size (7x7).
- ✓ A dropout layer to avoid overfitting.
- ✓ A MaxPooling MaxPool of size (2x2) allowing to reduce dimensions (weigh, high) of images issued of the previous layer after applying the different filters of Conv2
- ✓ A third convolutional layer Conv3 constituted of 32 filters of size (5x5).
- ✓ A dropout layer to avoid overfitting.
- ✓ A MaxPooling MaxPool of size (2x2) allowing to reduce dimensions (weigh, high) of images issued of the previous

layer after applying the different filters of Conv3

- ✓ A full connected layer FC allowing to transform the output of the previous layer into mono-dimensional vector.
- ✓ An output layer represented by a reduced mono-dimensional vector having as size the number of classes.

For all the previous layers a Relu activation function and a softmax function are used to normalize values obtained in each layer.

Table 1. Description of the Proposed CNN Model

Layer Type	Output Shape	Nb. parameters
conv2d_1 (Conv2D)	(None, 28, 28, 32)	1600
batch_normalization_1 (Batch)	(None, 28, 28, 32)	128
max_pooling2d_1 (MaxPooling2)	(None, 14, 14, 32)	0
conv2d_2 (Conv2D)	(None, 14, 14, 64)	100416
batch_normalization_2 (Batch)	(None, 14, 14, 64)	256
max_pooling2d_2 (MaxPooling2)	(None, 7, 7, 64)	0
conv2d_3 (Conv2D)	(None, 7, 7, 128)	401536
dropout_1 (Dropout)	(None, 7, 7, 128)	0
max_pooling2d_3 (MaxPooling2)	(None, 4, 4, 128)	0
batch_normalization_3 (Batch)	(None, 4, 4, 128)	512
flatten_1 (Flatten)	(None, 2048)	0
dense_1 (Dense)	(None, 128)	262272
dropout_4 (Dropout)	(None, 128)	0
batch_normalization_4 (Batch)	(None, 128)	512
dense_2 (Dense)	(None, 10)	1290
Total parameters		768,522
Trainable parameters		767,818
Non-trainable params		704

Table 2. Accuracy after applying different ML algorithms

Algorithm	Accuracy
LDA	86.41
KNN	96.90
CART	86.27
NB	57.04

Table 3. Loss value and accuracy value obtained when applying the proposed model

	Loss value	Accuracy value
Training set	0.059	99.86
Test set	0.633	98.29

Curves

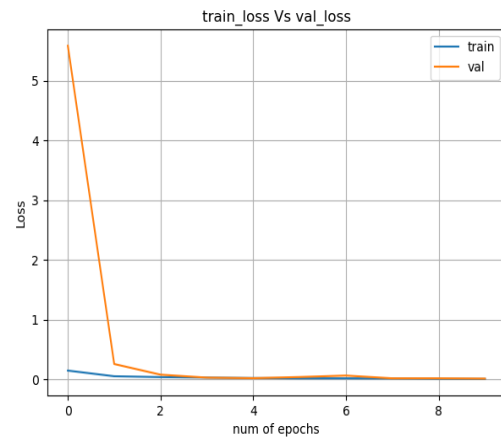


Fig 3. Training loss Vs Validation loss of the CNN model.

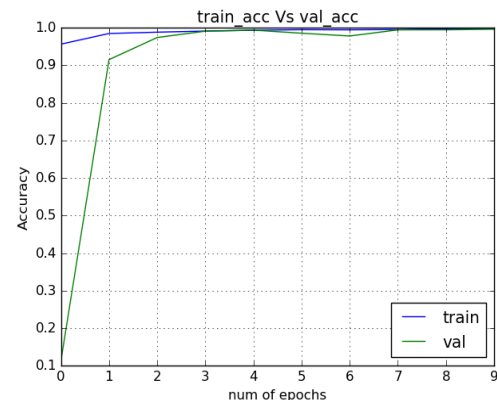


Fig 4. Training accuracy Vs Validation accuracy of the CNN model

6. Discussion

Table 2 summarizes the obtained results when applying the different ML algorithms including: LDA, KNN, DT (CART), NB. We observe that LDA, KNN and DT (CART variant) give a high classification accuracy (86.41, 96.90, 86.27), while and NB give a relatively low value of accuracy (57.04) compared to the previous algorithms. We note also, that the similarity metric used with KNN algorithm influence directly on algorithm performance.

Table 3 presents the obtained results when applying the proposed CNN model on the training set and the test set. Two performance measures are considered in this case, the loss value which calculates the sum of errors after

training the model, and the accuracy value which gives the rate of correctness.

In the same way, Fig 3 shows the evaluation of training loss and validation loss over time and in function of the number of epochs. It begins very high for the training set and ends very low because of the large number of samples, but its variation for the validation set is not very quick and appears relatively stable.

Similarly, Fig 4 plots the evolution of training accuracy and validation accuracy in function of the number of epochs. Contrary to the loss value, the accuracy starts very low and ends very high. This property is clearer with the training set because of its large size.

7. Comparison between ML and CNN approaches

In this section we try to establish a comparison between different algorithms, ML algorithms and the CNN model. The result of this comparison appears on table 4

Table 4. Comparison between ML approach and DL Approach

Algorithm	Accuracy rate
LDA	86.41
KNN	96.90
CART	86.27
NB	57.04
CNN model	99.86

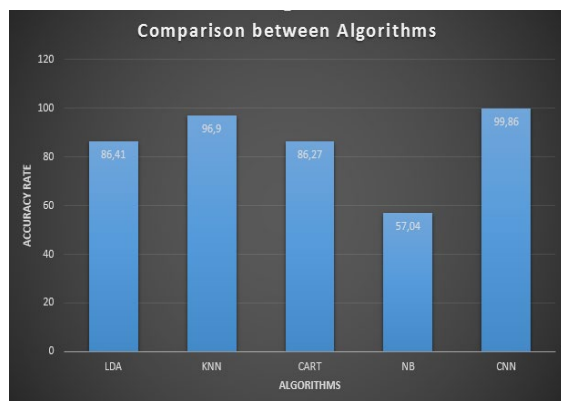


Fig 5. Comparison between different Algorithms

According to Table 4 and Fig 5, The comparison favors CNN model over ML algorithms.

8. Conclusion and Perspectives

In the last years, object recognition is based essentially on ML approach that gives a high performance. Few years later, some important

progress on ML area have been made especially with the apparition of a new sub-field called deep learning. It is mainly based on the use of many neural networks of simple interconnected units in order to extract meaningful patterns from a large amount of data to solve some complex problem such as: medical image classification, fraud detection, character recognition, etc. currently, we are able to use larger datasets to learn powerful models, and better techniques to avoid overfitting and underfitting. Until our days, the obtained results in this area of research are very surprising in different domains. We talk about very high values of accuracy which often exceed the threshold of 90%. For example, the accuracy rate on digits set is over 97%.

In the present paper, we have performed a task of classification on a clothing article dataset. We have used in a first stage many ML algorithms including: LR, LDA, KNN, DT (CART), NB, SVM. We obtained good result of accuracy especially on LR, LDA, KNN. In a second stage, we have built a CNN model to perform the same task of classification. The achieved performance is very surprising. We concluded our work by establishing a large comparison between different algorithms. The result of this comparison was in favor of DL approach through the CNN model we have built. As perspective of this promising work, we propose to improve these results by improving the architecture of CNN built model by changing some model parameters such as: the number of filters, the number of convolution and maxpooling layers, the size of each filter, the number of training epochs and the size of data batches. Another suggestion seems important, is to combine CNN with recurrent networks Resnets.

Référence :

- [1] Hinton, G. E., Osindero, S., and Teh, Y.-W.: A fast learning algorithm for deep belief nets. *Neural Computation*, 18, 1527–1554. (2006)
- [2] Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H.: Greedy layer-wise training of deep networks. In *NIPS'2006*. (2007)
- [3] Ranzato, M., Poultney, C., Chopra, S., and LeCun, Y. : Efficient learning of sparse representations with an energy-based model. In *NIPS'06*. (2007).
- [4] Jarrett, K. Kavukcuoglu, M. A. Ranzato, and Y. LeCun. : What is the best multi-stage architecture for object recognition? In *International Conference on Computer Vision*, pages 2146–2153. IEEE, (2009).
- [5] Krizhevsky. A.: Convolutional deep belief networks on cifar-10. Unpublished manuscript, (2010).
- [6] Y. Le Cun,, Y., Boser B., Denker J.S., Henderson D., Howard R.E., Hubbard W., Jackel L.D. :Handwritten digit recognition with a back-

- propagation network. In *Advances in neural information processing systems*, (1990).
- [7] LeCun Y., Huang F.J., and Bottou L.: Learning methods for generic object recognition with invariance to pose and lighting. In *Computer Vision and Pattern Recognition*, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, volume 2, pages II-97. IEEE, (2004).
 - [8] Lee H., Grosse R., Ranganath R. , and Ng A.Y.. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 609–616. ACM, (2009).
 - [9] Pinto N., Doukhan D., DiCarlo J.J. , and Cox D.D.: A high-throughput screening approach to discovering good forms of biologically inspired visual representation. *PLoS computational biology*, 5(11):e1000579, (2009).
 - [10] Turaga S.C., Murray J.F., Jain V. Roth F., Helmstaedter M., Briggman K., Denk W., and Seung H.S.. Convolutional networks can learn to generate affinity graphs for image segmentation. *Neural Computation*, 22(2):511–538, (2010).
 - [11] Abadi B., Agarwal M., Barham A, P, Brevdo E, Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M, Ghemawat S, Goodfellow I, Harp A, Irving G, Isard M, Jia Y, Jozefowicz R, Kaiser L, Kudlur M, Levenberg J, Mané D, Monga R, Moore S, Murray D, Olah C, Schuster M, Shlens J, Steiner B, Sutskever I, Talwar K, Tucker P, Vanhoucke V, Vasudevan V, Viégas F, Vinyals O, Warden P, Wattenberg M, Wicke M, Yu Y, Zheng X. TensorFlow: large-scale machine learning on heterogeneous systems 2015. <http://tensorflow.org/>. Accessed 1 Nov 2018.
 - [12] Theano Development Team. Theano: a Python framework for fast computation of mathematical expressions. *arXiv e-prints arXiv:1605.02688*. (2016)
 - [13] Chollet F, et al. Keras. 2015. <https://keras.io>. Accessed 1 Nov 2018.
 - [14] Paszke A, Gross S, Chintala S, Chanan G, Yang E, DeVito Z, Lin Z, Desmaison A, Antiga L, Lerer A. Automatic differentiation in pytorch. In: *NIPS-W*. 2017.
 - [15] Chetlur S, Woolley C, Vandermersch P, Cohen J, Tran J, Catanzaro B, Shelhamer, E. cudnn: Efficient primitives for deep learning 2014.
 - [16] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: *Neural information processing systems*. 2012. p. 25. <https://doi.org/10.1145/3065386>.
 - [17] Fukushima K. Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol Cybern*. 1980;36(4):193–202. <https://doi.org/10.1007/BF00344251>.
 - [18] LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. *Proc IEEE*. 1998;86(11):2278–324.
 - [19] Witten IH, Frank E, Hall MA, Pal CJ. Data mining, Fourth Edition: Practical machine learning tools and techniques. 4th ed. San Francisco: Morgan Kaufmann Publishers Inc.; 2016.
 - [20] Goodfellow I, Bengio Y, Courville A. Deep learning. Cambridge: The MIT Press; 2016.
 - [21] Minar MR, Naher J. Recent advances in deep learning: an overview. 2018. [arXiv:1807.08169](https://arxiv.org/abs/1807.08169).
 - [22] LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature*. 2015;521:436.
 - [23] Schmidhuber J. Deep learning in neural networks: an overview. *Neural Netw*. 2015;61:85–117.
 - [24] Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. *Nature*. 1986;323:533.
 - [25] LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, Jackel LD. Backpropagation applied to handwritten zip code recognition. *Neural Comput*. 1989;1(4):541–51. <https://doi.org/10.1162/neco.1989.1.4.541>.
 - [26] Hinton GE, Osindero S, Teh Y-W. A fast learning algorithm for deep belief nets. *Neural Comput*. 2006;18(7):1527–54. <https://doi.org/10.1162/neco.2006.18.7.1527>.
 - [27] Bengio Y, Lamblin P, Popovici D, Larochelle H. Greedy layer-wise training of deep networks. In: *Proceedings of the 19th international conference on neural information processing systems. NIPS'06*. MIT Press, Cambridge, MA, USA. 2006. p. 153–60. <http://dl.acm.org/citation.cfm?id=2976456.2976476>.
 - [28] Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, Berg AC, Fei-Fei L. ImageNet large scale visual recognition challenge. *Int J Comput Vision (IJCV)*. 2015;115(3):211–52. <https://doi.org/10.1007/s11263-015-0816-y>.
 - [29] Kumar M. An incorporation of artificial intelligence capabilities in cloud computing. *Int J Eng Comput Sci*. 2016. <https://doi.org/10.18535/ijecs/v5i11.63>.
 - [30] Saiyeda A, Mir MA. Cloud computing for deep learning analytics: a survey of current trends and challenges. *Int J Adv Res Comput Sci*. 2017;8(2):68–72. <https://doi.org/10.26483/ijarcs.v8i2.2931>.
 - [31] Dumbill E. What is big data? : an introduction to the big data landscape. 2012. <http://radar.oreilly.com/2012/01/what-is-big-data.html>
 - [32] Najafabadi MM, Villanustre F, Khoshgoftaar TM, Seliya N, Wald R, Muharemagic E. Deep learning applications and challenges in big data analytics. *J Big Data*. 2015;2(1):1. <https://doi.org/10.1186/s40537-014-0007-7>.
 - [33] Hinton G, Salakhutdinov R. Discovering binary codes for documents by learning deep generative models. *Top Cogn Sci*. 2011;3(1):74–91.
 - [34] Salakhutdinov R, Hinton G. Semantic hashing. *Int J Approx Reason*. 2009;50(7):969–78.
 - [35] <https://doi.org/10.1016/j.ijar.2008.11.006>.