Machine Learning on Cloud Systems



**Abstract**

It is an indisputable fact that technology has been grown up rapidly day by day. There are many areas of it those which require a lot of processing and demanding; such as areas like Big Data, Cloud Services, Data Science, Machine and Deep Learning etc. This paper will focus mainly on what are Cloud Services and Machine Learning, how Machine Learning can be applied to Cloud, which algorithms can be used in this manner, how system architecture should be, why Machine Learning and Cloud Systems are very important for future technology and development.

**1) Cloud Services and Machine Learning**

Basic definition for the Cloud Services is gathering or accessing and storing the data over Internet rather than using our computer’s hard-drive. In latest years, Cloud Services became more popular and one of the main reasons is demand for simplicity and accessibility for daily jobs of human life. As an illustration, accessing work documents on both at home and office without using USB or any other storage device. Nevertheless, accessing and storing are not the main properties of the Cloud Service. It should provide some functionality like virtualization, multi-tenancy, network access, elastic etc. [1]. Accessing and storing data should be fast and easy. In addition, elastic means we can perform those activities in anywhere at any time. Can we improve the efficiency of Cloud Computing? Or more specifically is there any development that we can combine with Cloud Computing? Perhaps one of the possible answers is nowadays world-wide development which is Machine Learning.

Machine Learning is quite promising expertise for the future; which is using in many technological areas. We can define it as “computers can learn without being programmed to perform specific tasks; researchers interested in artificial intelligence wanted to see if computers could learn from data” [2]. It has some similarities with Data Mining; both of them seek the data with their patterns. According to TechTarget the point that distinguishes them is Machine Learning uses data to detect pattern in data and adjust program actions accordingly [3]. We can understand the importance of Machine Learning by simply looking its area of usage. Several examples listed below:

* Face Detection
* Spam Filter
* Topic Spotting
* Spoken Language Understanding
* Medical Diagnose
* Customer Segmentation
* Weather Prediction

Machine Learning does not learn how to spam filter or how to detect face at the particular moment. It requires a lot of training data to learn the patterns and to identify the model; which means it requires a lot of resources where Cloud Services is also requires lots of resources. As an illustration, there is open-source MNIST database for developing the recognition of handwritings and identifying digits including over 60,000 examples [4]. If we would use this connection between Cloud and Machine Learning properly, we might convert the Cloud System to treasure for Machine Learning since it requires training examples in great quantities.

The essential part of Machine Learning is that we need to make sure what we are going to classify and out target. Another crucial point is that the relation between input and target should be predictable. To illustrate, if we consider to develop face recognition system with machine learning, we have to determine which parameters can influence the outcome, possible inputs and outcomes, system architecture, number of hidden layers and their continents, system decomposition of neural network, number and functions of perceptrons and sigmoid neurons etc. Paper will continue on this particular example which is Face Recognition System through Machine Learning.

**2) Face Recognition and Its Approach**

Most of the face recognition systems generally based on detecting and interpreting eye position, size, shape of eyes, nose, cheek bones by simple comparing these parts collected data with target data [5]. Developing face recognition system has some advantages like there is no need to test subject to work and it can perform mass identification. Nonetheless, it requires a lot of complexity and there are tremendous ingredients that we need to take into account. According to Dhiraj K. Das, computer scientist works in Nepal Engineering College, list of challenges represented as [6]:

* **Occlusion**: faces may be partially occluded by other objects
* **Presence or absence of structural components**: beards, mustaches and glasses
* **Facial expression**: face appearance is directly affected by a person's facial expression
* **Pose** (Out-of Plane Rotation): frontal, 45 degree, profile, upside down
* **Orientation** (In Plane Rotation):face appearance directly varies for different rotations about the camera's optical axis
* **Imaging conditions**: lighting (spectra, source distribution and intensity) and camera characteristics (sensor response, gain control, lenses), resolution
* **Facial feature extraction** (for local face recognition): To detect the presence and location of features such as eyes, nose, nostrils, eyebrow, mouth, lips, ears, etc. Usually assume that there is only one face in an image

Obviously developing face recognition system necessitates tremendous caution and complexity. Structural components like beard or jewel may decrease the success of system. Moreover, system should consider recognition in every aspect. There have been many developments using face recognition techniques and it is evolved through the years.

Generally it consists of 3 subtasks. Face Detection, feature extraction, recognition. Well known algorithms and methods are Eigenfaces, 3D-Recognition, Skin Texture Analysis, Thermal Cameras, Linear Discriminant Analysis (LDA), elastic bunch graph making using FisherFace, Hidden Makarov Model, Recognition with Controlled Background and Color [5]. We can simplify their manner of work by projecting human faces into multi-dimensional computer environment and extracting face proportions for comparison. Although many of methods works properly and accurately, actually their cost is high as well as time complexity due to multi-dimentional work way. Nevertheless, EigenFace method does not work in 3D dimension. It is primarily dimension reduction method which provides much better speed, efficiency and accuracy. Eigenfaces considered by many as one of the most convenient method to demonstrate minimum expected performance for system.

**3) EigenFace Algorithm for Feature Extraction**

EigenFace method created by Sirovich and Kirby, based on principal component analysis can be performed to collect human face images to form a set of fundamental features. More specifically, in 1990’s Kirby demonstrated that it is possible to represent human faces as a linear combination of weighted eigenvectors with usage of Principal Component Analysis (PCA) [7]. The aim is to obtain an image in lower dimension without losing too much information, feature of image and after then reconstructing it. More specifically, it represents a face in coordinate system. Since most of the humans have very similar geometry of face shape, this method aims to create two dimensional intensity pattern and recognition through statistical properties of human face. Both target and input images supposed to be on same pixel resolution. Nonetheless, Principal Component Analysis cannot obtain all variances of the image if we did not provide specific information in training data. To deal with it for instance, Wiskott offered another method that elastic bunch graph making for obtaining features of face images [8]. Other known offered methods are Kernel PCA, Independent Component Analysis (ICA) etc. In order to get better performance outcome, Jian Yang proposed 2DPCA [9]. It simply represents face images by matrices whereas PCA concatenate the image vectors. So their main difference is based on representing images on different scales. The advantages of 2DPDA is explained by Dhiraj K Das by: [10]

* By taking first largest eigenvalue and changing the number of training images, it is found that accuracy rate is high in the case of 2DPCA.
* It is found that if the difference between the eigenvalue is large, then the performance is same in all the cases but if the difference is small then recognition rate increases.
* The disadvantage of PCA method is computation time. It takes 2 hours to find the feature of image where in 2DPCA requires nearly 3-4 second to complete the same number of test images.

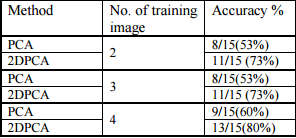


Figure 1: Results of Dhiraj’s experiment [10]

Therefore, for the classification of face recognition 2DPCA provides much more performance than PCA and we should use it in our architecture. Another crucial point that we can observe from the Figure 1 is if number of training image increases, the accuracy rate also increases; which training the machine is fundamental aspect of machine learning.

Later A.A Mohammed and his colleagues proposed bidirectional two dimensional principal component analysis (B2DPCA) for reducing the dimensionality of feature vectors [11]. It changes the face feature matrix with size N x N to M2 vector where M is a number. They observed that results are slightly better than 2DPDA. Its work way explained in below figure:

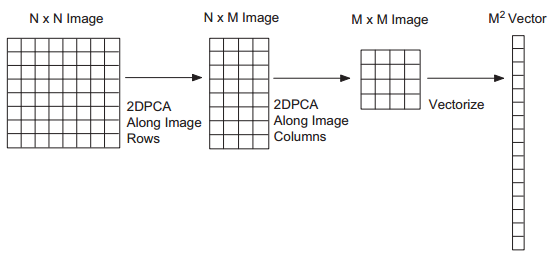


Figure 2: B2DPCA Block Diagram proposed by A.A Mohammed [11]

**3.1 How the EigenFace Method Works?**

Steps of EigenFace methods listed below

1. The Eigenvectors calculated by face image and k-number of eigenvectors
2. Highest Eigen value is chosen.
3. Calculate Euclidean Distance by weight vectors generated from detected face by projecting them onto Eigen spaces. Also *Nearest Neighbor* algorithm can be used to find Euclidean distance between two vectors.
4. If minimum Euclidean distance < predefined threshold then it is same person.

**4) Locate and Detect Head**

Locating and detecting head is essential before we analyze and compare the head. It can be done by motion detection and several tracking algorithms. Matthew Turk and Alex Pentland proposed article “EigenFaces for Recognition” which indicates that with usage of simple spatiotemporal filtering (e.g. frame differencing) emphasize image locations change in entire image regarding in time [12]. Moving persons “lights up” in filtered image. After locating persons, method analyze “motion blobs” in time to decide whether this motion caused by person or not [12]. Some of the rules are “the head is small upper blob above a larger blob (the body)”, “head motion must be reasonably slow and contiguous”. Matthew Turk and Alex Pentland indicated entire system in figure below

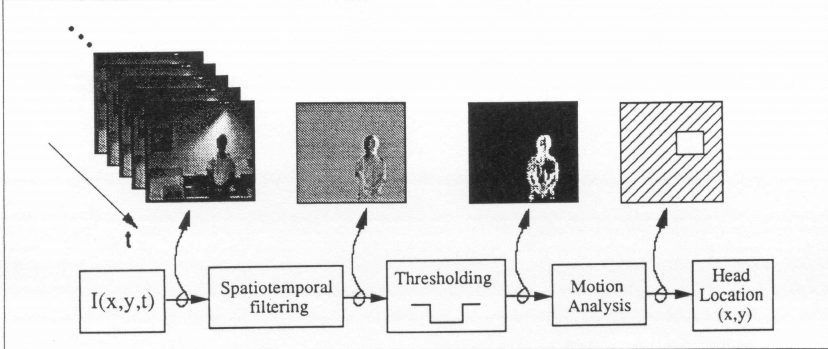


Figure 3: Head Tracking and Locating System [12]

**4.1 Detecting New Faces**

Detecting or learning new faces can be interpreted as unsupervised learning of machine learning. If input face is close enough to face space but not classified as “known face”, image labeled as “unknown face” [12]. Turk and Pentland proposed that pattern vectors in the clusters checked for similarity. If the image will pass similarity test, the average of the feature vectors will be added in database as “known face” [12].

**4.2 Interpreting Detected Face as Human Face or Not**

We shortly explained how to detect the face image from entire image however it may not be human face at all. Our system should include the feature to decide captured image is human face or not. Since machine learning is based on work way of human neural network, system has to include some input neurons, middle layers (hidden layers) and output neuron(s). Input neurons are our detected face images, hidden layers are the functions that do the identification. Lastly output neuron(s) simply states it is human face or not, in our case one output neuron is enough. If our system use 28 x 28 pixel resolution, then input layer should include 784 (28x28) neurons. Michael A. Nielsen explained this in figure below

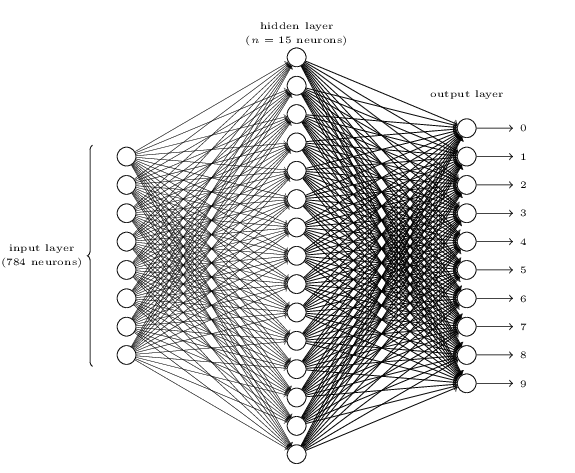


Figure 4: Explanation of Neural Network System [13]

To decide detected object is a face or not, our hidden layers should include properties that distinguish human faces from other faces like animals or meaningless objects. When input comes from input layers, hidden layers should interpret these data. To do so, specific algorithms can be applied to hidden layers. Michael A. Nielsen indicated this as below:

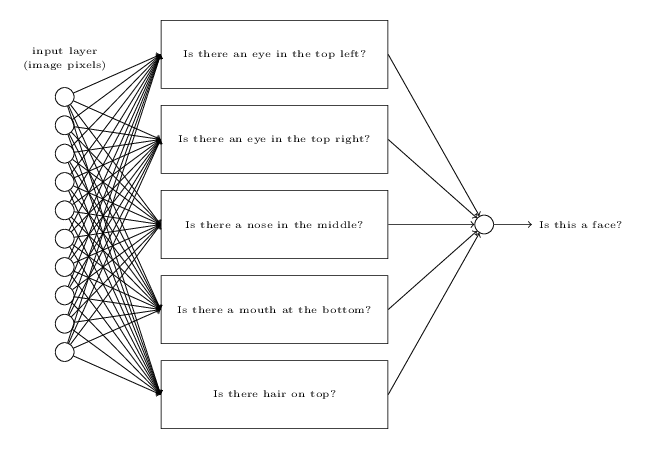


Figure 5: Hidden Layers Functionalities [13]

**4.3 Training Examples for Machine Learning**

There are some precious databases for training example of face recognition. For example, there is a Facial Recognition Technology (FERET) database used by Thai Hoang Li, Len Bui with their research on “Face Recognition Based on SVM and 2DPCA”. It consists of 14,051 approximate images with eight bit scale images [14]. Another one is that AT & T database of Cambridge University, the size of each image is 92x112 pixels [15].

**4.4 Block Diagram of Entire System**

With the work way of Matthew Turk and Alex Pentland spatiotemporal filtering for detecting face position in the image, B2DPCA classification of A.A Mohammed and his colleagues, block diagram of “Face Recognition in Cloud with Machine Learning” created as below:



Figure 6: Block Diagram

**REFERENCES**

[1] – Cloud Computing Website

<http://www.pcmag.com/article2/0,2817,2372163,00.asp>

[2] – Machine Learning and Its Importance

<https://www.sas.com/en_us/insights/analytics/machine-learning.html>

[3] – Machine Learning on TechTarget

<http://whatis.techtarget.com/definition/machine-learning>

[4] – MNIST Database for Handwritings

<http://yann.lecun.com/exdb/mnist/>

[5] – Face Recognition Wikipedia

<https://en.wikipedia.org/wiki/Facial_recognition_system>

[6] Comparative Analysis of PCA and 2DPCA in Face Recognition, Dhiraj K. Das

<http://www.ijetae.com/files/Volume2Issue1/IJETAE_0112_57.pdf>

[7] M. Kirby, L. Sirovich, Application of the Karhunen–Loeve procedure for the characterization of human faces, IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (1) (1990) 103–108.

[8] L. Wiskott, et al., "Face recognition by elastic bunch graph matching," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 19, pp. 775-779, 1997.

[9] Y. Jian, et al., "Two-dimensional PCA: a new approach to appearance-based face representation and recognition," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 26, pp. 131-137, 2004.

[10] Comparative Analysis of PCA and 2DPCA in Face Recognition, Dhiraj K. Das

<http://www.ijetae.com/files/Volume2Issue1/IJETAE_0112_57.pdf>

[11] - Human face recognition based on multidimensional PCA and extreme learning machine

<http://ac.els-cdn.com/S0031320311000987/1-s2.0-S0031320311000987-main.pdf?_tid=4f3d8466-fe56-11e6-a1e1-00000aab0f01&acdnat=1488355913_f52b4a3a335f77f3a3f1a65a1ae7659a>

[12] – Matthew Turk and Alex Pentland - Eigenfaces for Recognition

<http://www.face-rec.org/algorithms/PCA/jcn.pdf>

[13] – Michael A. Nielsen, “Neural Network and Deep Learning”, Determination Press, 2015

[14] – FERET Database for Faces

<http://www.itl.nist.gov/iad/humanid/feret/feret_master.html>

[15] – Cambridge University AT&T Database for faces

<http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>