Comprehensive Analysis of Machine Learning Algorithms for Face Detection

Bhavana, V. Jagan Naveen, K. Krishna Kishore

Abstract*: *Face detection is the most common application used in security system, cameras, fun face filter apps, etc. many techniques and algorithms are introduced by developers for face detection in real time but all techniques or algorithms does not give best results while applying on all ranges of processors. In this, three machine learning algorithms i.e. Histogram of Oriented Gradient, Haar cascade classifier and deep neural networks implemented on different processors for verifying processing speed of each algorithm on the different processor.

Keywords* : *Deep neural network, Haar cascade classifier, Histogram of oriented gradient, Machine learning.

I. INTRODUCTION

Now a day's machine learning algorithms play a key role in different security systems like cyber security, home security, military, etc. For the home security system, the system is a combination of controllers, sensors, video cameras, and software (machine learning techniques) [1] that give a smart home security system. Machine learning techniques used for face detection and recognition, object detection, speech recognition and smoke detection in home security system. Machine learning algorithms are three types supervised, unsupervised and reinforcement learning [2], and common algorithms are support vector machine, ada boost, neural network, etc. There are many machine learning algorithms like Histogram of Oriented Gradient [3], Haar Cascade Classifier [4] and Deep neural network [5] used for face detection. The mentioned algorithms implemented on both Raspberry Pi 3 board and PC and compared on parameters elapsed time and face detection accuracy.

II. HISTOGRAM OF ORIENTED GRADIENT

Histogram of Oriented Gradient used for feature extraction from the image where feature extraction helps to retrieve useful information from the image. For feature extraction HOG vertical and horizontal gradient calculated of gray scale image using a mask (-1, 0, 1) horizontally and vertically [3]. Later magnitude and direction of the gradient calculated using equation 1 and equation 2

$$g = \sqrt{f_x^2 + f_y^2} \qquad \text{Eq. 1}$$

$$\theta = \arctan\left(\frac{gy}{gx}\right) \qquad \text{Eq. 2}$$

Using the values gradient drawn from high gray scale value to a low gray scale value. Repeating same process for all the pixels in the image a gradient image that removes non-essential information from the image. Later the image subdivided into 16 × 16 cell gradients of each cell calculated and replaced. The HOG pattern image compared with trained data for extracting information using linear SVM.

Manuscript published on 30 August 2019.
*Correspondence Author(s)
Bhavana, ECE, GMR Institution of Technology, Rajam, INDIA.
V. Jagan Naveen, ECE, GMR Institution of Technology, INDIA.
K. Krishna Kishore, ECE, GMR Institution of Technology, Rajam, INDIA

III. HAAR CASCADE CLASSIFIER

Haar wavelet is the main base for the haar cascade classifier these wavelets are square shaped sequence of rescaled functions. From the haar wavelet function, haar-like features derived. Haar-like features are adjacent rectangles that contain two region black regions and white regions in different patterns [4]. These features are convolution kernel applied on gray scale image for detection of a human face in the image.

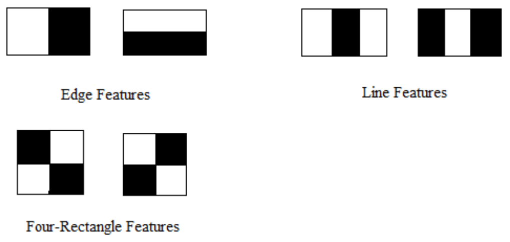


Figure 1: Haar Features

Let us consider a gray scale image that contains human face. To detect nose bridge region in the image, we consider a three adjacent rectangle edge feature with black region and white region in the middle as nose bridge region is brighter than eyes. In this manner, each individual edge feature matches a particular feature on the face. We place three adjacent rectangular features on the nose bridge region such that the middle white region. Then calculate the difference between the pixel's value under the white region and the black region. The result is a round off to 0 or 1, so face feature is a nose or not.

Initially, haar like features are single pixels they can extend to over 16000+ haar features. Using all features for detecting face, it take a long time and a few not used for face detection for customizing the process integrate image, Ada Boost and a cascade used.

Integrate image derived from the original image by adding all left pixel values, all top pixel values and its values and assigning to itself.

$$ii(x,y) = \sum_{x' \leq x, y' \leq y} i(x', y') \qquad \text{Eq.3}$$

Integrate image used to reduce the number of operations performed using a normal image. Below example shows how integrate image reduces the number of operation then normal image.

Table 1: Original Image Pixel Values

3	4	5
1	9	3
0	6	9

To calculating average of pixel matrix, we have to performed 9 operations as:

(3+4+5+1+9+3+0+6+9)/9 = 4.44

Integrate image has pixel values as shown in table 2. For example we want to calculate the pixel value of (2,2) pixel then it will be the sum of (1,1), (1,2), (2,1) and (2,2) and the sum value replaced with pixel values.

Table 2: Integrated Image Pixel Values

3	7	12
4	17	25
4	23	40

This integrated image will reduce the number of operations as compared to the original image. For example, if we need to calculate the average that we just need to pick the last pixel value and divide it by the number of values.

Ada Boost adds all weak classifiers and forms a strong classifier by assigning weights to a weak classifier [6]. Weak classifiers are edge features that can guess over 50 images out of 100 images containing face. They assign these weak classifier with weights i.e. if classifier reaches its trained threshold value while calculating pixels value under the feature, then its weight is 1 or else 0.

$F(x) = \alpha_1 f_1(x) + \alpha_2 f_2(x) + \alpha_3 f_3(x) + \ldots\ldots\ldots$ Eq. 4

F(x) = strong feature

α = weight of weak feature

f(x) = weak feature

Cascading helps to arrange all classifier into stages that avoids testing all 6000 edge features on an image. They divide classifiers into N stages, each stage contains individual classifiers if the image satisfies the stage1 then image is send to stage 2, else will not pass to stage 2. Below diagram shows the function of cascade.

IV. DEEP NEURAL NETWORK

Deep learning is sub concept of machine learning, which involves with deep neural network for image classification. Deep neural network are typical conventional neural networks with more number of hidden layers [5].

Neural network or artificial neural networks are designed from the inspiration of biological neural network, mostly used for solving artificial intelligence problems and developing artificial intelligence. Neural network formed from neurons these neurons are connected to each other with some weights within the neuron some math is performed i.e. sigmoid function or logistic function or hyperbola tangent. A simple architecture of neural network is organization of neurons in layers where neuron in each layer connected to every neuron on other layer as shown in Figure

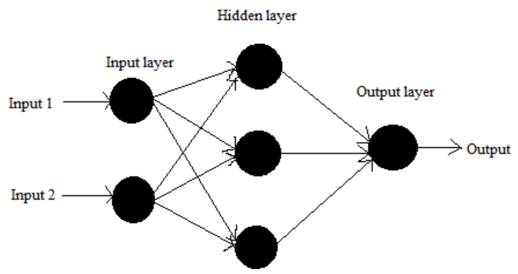


Figure 2: Simple Neural Network

The above image is a simple neural network but for dealing with huge data in the form of multiple array ConvNets developed [7]. Convolution Networks Network contains multiple hidden layers where each layer performs certain tasks.

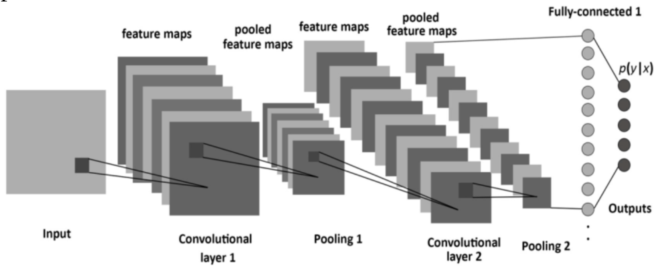


Figure 3: Convolution Neural Network

For any real time implementation of the Convolution neural network the network trained and tested. Developers and researches has designed many Architectures and Frameworks of Deep Neural Network for real time implementation like AlexNet, VGG, GoogLeNet, ResNet, Inception series (V2, V3 and V4) , MobileNet, SqueezNet [8], caffe [9], etc .

V. HARDWARE AND SOFTWARE REQUIREMENT

Raspberry Pi 3runs on Broadcom BCM2387 chipset of Quad-core ARM cortex-A53 processor at speed of 1.2 GHz, with RAM of 1.0 GB. Laptop runs on Intel (R) Core (TM) i5-3337U central processing unit at speed of 1.80 GHz, with RAM of 4.00 GB. Web camera is a USB video camera that streams images to system in real time. Raspbian OS is the operating system for Raspberry Pi. Libraries used Open CV and other supporting libraries. For deep neural network application Mobile Net architecture is used with caffe framework.

VI. RESULTS

The three techniques implemented in real time for face detection in live streaming on both Raspberry Pi and laptop. While implementing on laptop, laptop's web cam is used for accessing live streaming.

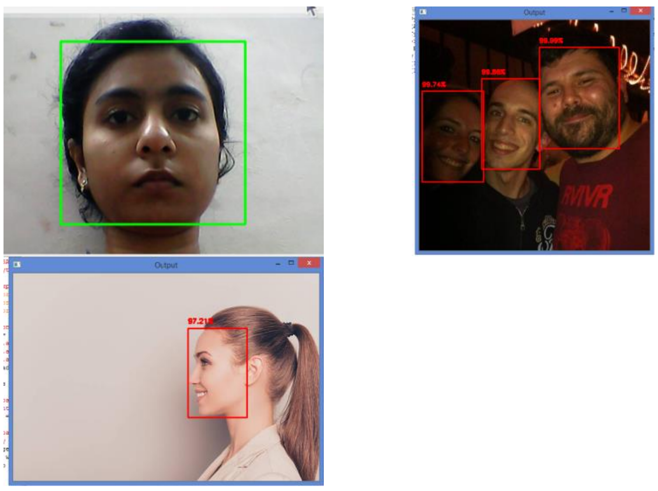


Figure 4: Screenshots of face detected

Deep neural network is more accurate, compare to other two techniques in detecting face as shown in above figure it can detect the face from any angle it can also detect half face and face in dim light. This technique takes 3.4 seconds for initializing the architecture, which are trained for face detection. Whereas Haar Cascade is not as accurate as Deep Learning but better than HOG and it takes 1.4 seconds to initialize its trained cascade. The minimum and maximum elapsed time taken by the techniques tabulated.

Table 3: Elapsed Taken by Techniques on PC

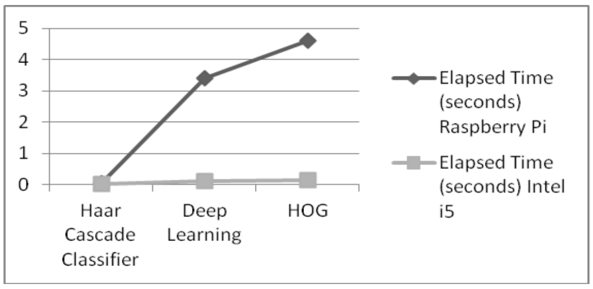
Techniques	Elapsed Time (seconds)	
	Minimum	Maximum
Haar Cascade Classifier	0.015	0.031
Deep Learning	0.109	0.156
HOG	0.4	0.5

On Raspberry Pi Haar Cascade Classifier gives the best accuracy compared to other techniques. Where Haar Cascade Classifier takes initialization time between the ranges of 1.3—1.8 seconds and it takes 0.05 seconds when the face is not present in frame and 0.1 seconds for detecting face if the face is present in the frame.

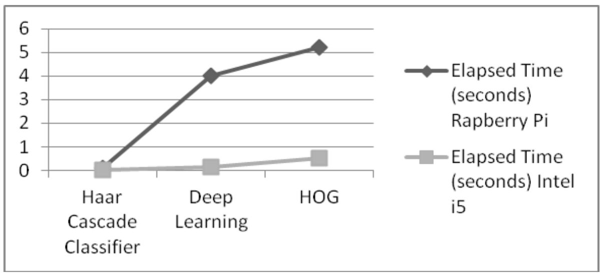
Table 4: Elapsed Time Taken by Techniques on Raspberry Pi

Techniques	Elapsed Time (seconds)	
	Minimum	Maximum
Haar Cascade Classifier	0.05	0.1
Deep Learning	3.4	4.0
HOG	4.6	5.2

The Graphs show the minimum and maximum time taken by techniques on both Raspberry Pi 3 and PC. The behavior of techniques varies by processor they run on. Haar cascade classifier does not differ by changing processors whereas deep learning shows much difference in processing time, while dealing with images deep learning algorithm gives best results while it run on GPU (graphical processing unit) [10].



Graph 1: Minimum Elapsed Time Taken by Techniques



Graph 2: Maximum Elapsed Time Taken by Techniques .

VII. CONCLUSION

In this paper machine learning techniques HOG, Haar cascade classifier and deep learning implemented on both PC and Raspberry Pi 3. Elapsed time taken by same techniques are different on both Raspberry Pi 3 and PC. The performance of a security system depends on the techniques and the processor used. As Raspberry Pi has RAM of 1GB, if on that we use complex architecture, which occupies storage of 300 MB of RAM then the technique consume more elapsed time, that causes delay. The Haar cascade classifier takes less time compared to other techniques for face detection. Deep learning techniques have a complex architecture deep neural network that occupies more space to overcome.

REFERENCES

1. G K, Vasyl T. Structure and model of the smart house security system using machine learning methods. In2017 2nd International Conference on Advanced Information and Communication Technologies (AICT) 2017 Jul 4 (pp. 105-108). IEEE
2. "Machine Learning For Dummies" by John Paul Mueller and Luca Massaron
3. Han F, Shan Y, Cekander R, Sawhney HS, Kumar R. A two-stage approach to people and vehicle detection with hog-based svm. InPerformance Metrics for Intelligent Systems 2006 Workshop 2006 Aug 21 (pp. 133-140).
4. Viola P, Jones M. Rapid object detection using a boosted cascade of simple features. CVPR (1). 2001 Dec 8;1(511-518):3.
5. Farfade, S.S., Saberian, M.J. and Li, L.J., 2015, June. Multi-view face detection using deep convolutional neural networks. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval* (pp. 643-650). ACM.
6. Freund Y, Schapire RE. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of computer and system sciences. 1997Aug 1;55(1):119-39.
7. LeCun Y, Bengio Y, Hinton G. Deep learning. nature. 2015 May;521(7553):436
8. Indola FN, Han S, Moskewicz MW, Ashraf K, Dally WJ, Keutzer K. SqueezeNet:AlexNet-level accuracy with 50x fewer parameters and <0.5MBJmodel size. arXiv preprint arXiv:1602.07360.2016 Feb 24

9. Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S, Darrell T. Caffe: Convolutional architecture for fast feature embedding. InProceedings of the 22nd ACM international conference on Multimedia 2014 Nov 3 (pp. 675-678). ACM
10. “Artificial intelligence computing leadership from Nvidia”, www.nvidia.com

AUTHORS PROFILE

Bhavana received her Bachelor of Technology in computer Science From Andhra University College of Engineering. Her area of interest is embedded systems.

Dr. V. Jagan Naveen received his Ph.D. from Andhra University. He is currently working as Professor in ECE department at GMRIT, Rajam, India. He published 25 papers in National and International Journals/Conferences. His area of interest is signal processing and wireless communication.

K. Krishna Kishore received his Bachelor of Technology and Master of Technology from JNTU, Hyderabad. He is working as Assistant Professor in department of ECE, GMRIT, Rajam, India. His area of interest is wireless communication.