ISSN 2347 - 3983
Volume 8. No. 3, March 2020
International Journal of Emerging Trends in Engineering Research
Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter29832020.pdf
https://doi.org/10.30534/ijeter/2020/29832020

Heuristics of Machine Learning on Embedded Intelligence Device for Multiple Object Detection

Navya Sneha C[1], Satyanarayana P[2]
[1]Mtech Student, KLEF, Andhra Pradesh, India.
[2]Professor, Department of ECM, KLEF, Andhra Pradesh, India

ABSTRACT

The application of Machine Learning algorithms in the embedded intelligence device is still a challenge to many researchers. This can be achieved by implementing Machine Learning algorithms in devices like computers or mini-computers and later this can be extended to high performance embedded intelligent boards. In this project the primary focus will be on how the Machine learning algorithms provide the data processing in the given data sets to provide an optimal solution. In this way the motion of objects pertaining to an area can be easily tracked and identified. If any intrusion occurs then it can be easily detected without any further delay. For this process to happen the various aspects are a cloud based server which the embedded intelligence device is connected, a suitable and efficient Machine learning algorithm and lastly inclusion of the required data sets for the detection or we can use the data sets that are already present in the open cloud server.

Key words: Convolutional Neural Networks (CNN), Deep Learning (DL), Embedded systems (ES), Histogram of Oriented Gradients (HOG), Machine learning (ML), Support Vector Machine (SVM).

1. INTRODUCTION

Machine learning is a series of various techniques that learn from their experience by analyzing the data and renewing the models on how previous actions take place and using those insights to make better choices in the future [8]. Such approaches are used in an orderly manner to assess how present conditions impact communications performance and automatically pick the arrangement or configuration for better accomplishment, even in highly dynamic missions. To pick a configuration in real-time, the domain requires the decision-maker's capacity within the radio stack and IP stack decision-making chain. The SVMs are conscious of two communication networks and general-purpose processor, whereas the current SVM is concentrated in memory and libraries are delayed. There are few methods to customize the current SVM library to boost runtime 20 x faster and the device regulates memory footprint. On the other hand, smart devices are used to independently process the data, evaluate the situation and act [6]. For example, the user remotely controls the mobile application for thermostat, but based on the occupancy of home and user comfort, a smart thermostat will automatically adjust the heat [7].

The fundamental multi-disciplinary of machine learning has only been recognized in the past decade or so. To make computers understand, it fuses concepts from neuroscience and genetics, statistics, mathematics, and physics. One aspect that has guided the move towards machine learning development is data mining, that aims at extracting information that is useful from large datasets (by individuals with devices and wallet protectors) and needs powerful algorithms, developing more emphasis on computer science. We will also be interested in computational complexity of machine learning techniques, since we generate algorithms. It is extremely important that we would like to use any of the methods over very large amounts of data, so it will be a concern for algorithms that have high degree polynomial sophistication in the datasets size. The complexity is often broken down into two parts: training complexity, and the complexity of implementing the learned algorithm. Training will not happen quite often, usually is crucial in practice, because it can take more time. Nonetheless, we always want a quick decision about such a testing point, and if an algorithm is being used there are possibly lots of test points, so this has to be lower computational cost.

The tasks like object detection, classification and tracking [13] will be required for so many real time applications of an IoT architecture's perception or sensor layer [1]. These attributes depend on a huge amount of data and involve quick response, therefore, the scene should be seized and stored, so that decision making is possible in real time. Determining in real time as live performance [14], grasped as ability to solve an application issue using only the accessible frame from a live video feed at any time without collecting immediate frames for processing delay, this prerequisite can never easily be encountered by cloud computing due to throughput and the possibility of limited coverage. In these instances, at network edges there is possibility of optimization, i.e., IoT sensor layer, leading to what would be edge computing [2]. For getting accuracy of such visual sensing endpoints has been one of main criteria for the architecture, as connected IoT infrastructure devices have to be powered by batteries in certain scenarios, such as smartphone and outdoor applications. Therefore, they may be self-powered by, for example, harvesting techniques which include energy. Image processing approaches vary from conventional computer vision algorithms towards more modern use of deep learning techniques. The potential of above has led to new phase of image processing in recent years [3].

Neural networks are two types one of them is known as CNNs (Convolutional Neural Networks) [4], which when combined together with recent computing abilities, propose a new approach for solving tasks of computer vision. Amongst benefits of them are compared to standard computer vision algorithms were their reliability and precision. While conventional algorithms are designed for specific purposes and situations, they have been trained in a significant way to solve very general issues. Even though the training phase is typically very efficient in terms of computation, the advantages of them could be utilized using less complex[12] hardware resources when trained. The findings of the last few years show an optimistic future of image processing using them. The emphasis was for a range of applications such as segmentation [6], image recognition, object detection [5] and object tracking [7]. Considering successful results across most of them, deep-featured object tracking and they have only recently emerged. One of it's state-of - the-art criteria for tracking of an object is the visualization challenge [8] as well as the results of past years have indeed been focused on both deep features [11] and deep learning techniques [9]. The developing and challenging of algorithms and CNNs endlessly to become more accurate and quicker but are typically evaluated to benchmark databases [10] on strong hardware platforms. At the other side, designers and researchers are looking to diversify the hardware available options to provide all resources required for implementation to improve efficiency of such challenging networks. Nevertheless, there may be a shortage of IoT end-nodes for processing communication and image capture.

2. MACHINE LEARNING IN EMBEDDED SYSTEMS

For many inferencing tasks, general-purpose processors are not required, so researchers found that GPUs provide full resolution of floating-point arithmetic therefore network pruning uses post-training DNN analysis to eliminate connections between neurons that have little or no effect on performance [10]. The needless calculations and, as an additional benefit, significantly reduce the energy consumed by memory accesses due to the deletion of their connections.

The field-programmable array (FPGA) is well positioned to take advantage of approximate-computing and similar enhancements. Most of the implementations of the ARM and other microprocessors now have single-instruction multiple-data (SIMD) pipelines that are designed for low-resolution integer data, up to 8bit, and the FPGA allows for fine-grained bit resolution. Xilinx provides a reduction of precision from 8-bit to binary that reduces the use of FPGA lookup table (LUT) by 40 times.

The platforms like the Ultra96board make this possible for separating processing across a general-purpose processor and a programmable-logic fabric with support for close interactions between the two. Such a platform allows the fast deployment of complex DNN architectures based on the Xilinx Zynq UltraScale+ MPSoC architecture to enable use of special purpose layers. To handle convolutional and pooling layers, as well as thresholding functions, while the Zynq ARM Processor, running a Linux-based operating system and supporting code, handles data flow and executes layers with less predictable memory access patterns that we use the FPGA fabric within the MPSoC system.

The designers of the Ultra96board used the programmable DSP slices, which can be conveniently programmed to serve as dual 8bit integer processors, as well as the search table logic for implementing binary or ternary processing within a platform's FPGA fabric. The key advantage of using programmable hardware is the ability to perform layer mixing. Without using external memory as a buffer, the arithmetic units can be connected directly or through block RAMs in order to forward data generated by each layer. Although machine learning algorithms can be complex to build, the research community has adopted principles such as open source and the result is that there are a range of freely available packages that developers can call on to implement systems, with the exception that most of them have been created for database platforms. To port DNN implementations to FPGA for users on their hardware vendors such as Xilinx have made it possible. The Xilinx ML Suite, for instance, can take a Caffe DNN[13] template that has been trained in the desktop or server environment and port it to a Zynq-based target, and the translator will break processing between custom software compiled for the arm processor and a pre-optimized neural convolution network loaded into the FPGA fabric.

3. OBJECT DETECTION APPROACH

For the detection of objects in a particular area we can use the following machine learning techniques: Feature-based algorithm (HOG + Linear SVM); YOLO (You Only Look Once); Deep Learning based algorithm (SSD + MobileNets). In this we have used all the three approaches and calculated Mean Average Precision (mAP). We used the implementation of algorithms using OpenCV because it has pre-defined functions for Computer Vision applications. The following explains usage of the above algorithms.

3.1 Feature-based algorithm (HOG + Linear SVM)

OpenCV comes with a pre-trained HOG + Linear SVM framework that can be used to detect objects in both images and video streams. This system is based on Dalal and Triggs method to automatically detect persons in images or video streams. Whereas the HOG approach appears to be even more accurate than in its Haar counter-part, this still demands that somehow the parameters for detectMultiScale be properly set.

The above approach will be as follows for object detection: First from training set (positive) we need to extract HOG features; HOG vectors with features need to be computed from training set (negative); Linear SVM training should be done; Hard-negative mining of the data should be applied; Using negative samples, positive samples and hard-negative samples we need to re-train your Linear SVM; Now object detector can be applied to our dataset.

3.2 YOLO (You Only Look Once)

To help improve the speed of deep learning object detectors, we use YOLO as a one-stage detector strategy The algorithm approaches object detection as a regression problem, taking a provided input image and simultaneously learns bounding box coordinates and associated label class probability. Farhadi and Redmon are in a position to accomplish such a large number of object detections by conducting joint training to both object detection and classification. Utilizing joint training, the researchers trained YOLO9000 concurrently on both the ImageNet dataset and the COCO dataset. With the help of above training datsets we obtain results for both images and video streams.

3.3 Deep Learning based algorithm (SSD + MobileNets)

The methods used for the Real Time Object Detection are Single Shot Detectors (SSD) and MobileNets. We can obtain superfast real-time object detection on resource constrained devices (including the Raspberry Pi, smartphones, etc.) when these methods are combined together. For this application we are using a deep learning framework called CAFFE developed by the Berkeley Vision and Learning Center (BVLC). It is written in C++ and has Python and Matlab bindings. It has four steps:

Step 1 - Data preparation
Step 2 - Model definition (.prototxt)
Step 3 - Solver definition (.prototxt)
Step 4 - Model training (.caffemodel)

The MobileNets use Depthwise seperable convolution which uses the standard convolution layer and a point wise convolution for the processing of the data in the frames. For this the operation cost will be

$D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F$

Where D_K is the Kernel size,
M is the Number of input channels,
N is the Number of output channels and
D_F is the Feature map size.

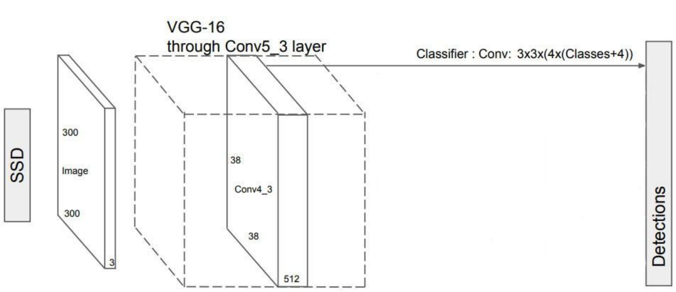

Figure 1: SSD Architecture

While the architecture of SSD (Single Shot Detectors) in figure 1 will have VGG-16 convolutional layer and a classifier that gives the number of detections in an image frame. The calculation for the bounding boxes for the object detections is as; Let us consider Conv4_3 layer. It is of size 38*38*512; 3*3 convolution applied; 4 bounding boxes. Therefore the output will be 38*38*4*(c+4). If c=21 (i.e., 20 object classes +1 background class), the output will be 38*38*4*(21+4) =144,400. In terms of bounding boxes, there will be 38*38*4=5776bounding boxes.

4. IMPLEMENTATION OF OBJECT DETECTION MODEL

For the hardware we used the PC with high speed processor and ability to take live video feed for the detection of objects at any particular point of time. The software requirements include OpenCV installed in the PC which is able to compute high speed applications. To give a brief on how the object detection works the figure 2 describes about the proposed system. Here we will be having the input block which capture the images and videos and feed them to the processor which

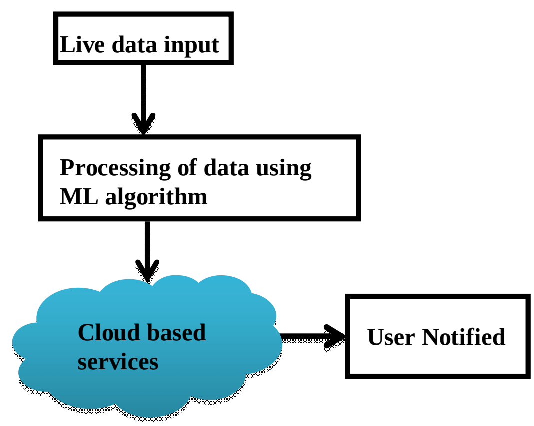

Figure 2: Proposed System for object detection

have pre-trained ML algorithms and there the detections of objects will take place. Next we sent the detection to the cloud based services wherein they store the data and gives the user notified by SMS/e-mails. This system is more compatible with the Linux platforms so the object detections will take with more accuracy. And also for the PC operating on windows we need to have supporting tools that improve its ability to high end applications. This project is done on PC with implementation in ANACONDA software. It is very great tool that supports and implements the high range computer vision applications with speed and accuracy.

5. RESULTS

For the detection persons using Feature –based algorithm there encountered a problem related to person detection. We tend to observe that there are many bounding boxes related to person because it identified human parts differently rather than a single parameter. So with suitable adjustment in the detectMultiScale parameter we can reduce this problem. The table 1 shows the detections and accuracy when we performed using YOLO algorithm. Here we tend to get accuracy more with increase in number of objects in the frame. But this algorithm fails to provide that accuracy in the live streams. This problem is overcome in Deep learning-based algorithm approach where in the accuracy tend to maintain same with the increase in objects in a frame. The table 2 illustrates the same with objects and accuracy/precision.

Table 1: YOLO based object detection results

Number of Objects	FPS (Frames per second)	mAP (mean average precision)
1	2.6230	72.03
2	1.7179	94-95
3	1.6987	96-98
4	1.8808	91-98
5	1.8640	56-99
6	1.6829	64-99
7	1.6952	56-98
8	1.8389	75-99
9	1.8364	80-99
>10	**2.1008**	**98-99**

The comparison of algorithms is shown in table 3. While it is observed that the deep learning algorithm provided better results in terms of accuracy for live video streams, we used this algorithm for the home intrusion system. Whenever there is any unknown person has entered the house then system will capture the image of that person and sent it to cloud based services along with an alert message.

Table 2: Deep Learning based algorithm results

Number of Objects	FPS (Frames per second)	mAP (mean average precision)
1	6.87	80-98
2-3	6.90	95-99
4-n	**6.93**	**98-99**

Here we have used the AWS (Amazon Web Services) because it is very flexible and has S3 database for storing the data so we can gain access to that from anywhere. For sending the alerts we used Twilio because the API provides the accuracy without any delay. In addition to the results obtained using parameters the objects detected in a particular window are shown in figures 4, 5. Figure 4 shows the YOLO algorithm output in which different objects are identified because the dataset we used is COCO which has 80 in-built object labels. While in figure 5 is obtained using deep learning-based algorithm which has 20 object class + 1 background class to detect the objects in a frame. The dataset used contains CAFFE framework. The R-CNN are typically comes under the traditional ML algorithms and they are very time consuming. It had low mAP (Mean Average Precision) of 73.2% at only 7 FPS (Frames per Second). To implement the R-CNN we need two shots one for regional proposals generation and other for detecting the object in each proposal.

Table 3: Comparision results of algorithms used

Algorithm	Time elasped	mAP
Feature-based(HOG+Linear SVM)	>3 seconds	50-60
YOLO	3-4 seconds	80-90
Deep Learning Algorithm (MobileNets+SSD)	>4 seconds	>90

YOLO performs 63.4% mAP at 45 FPS which is better than R-CNN. The Single Shot Detector (SSD) together with MobileNets provide very accurate and achieves 77.2% mAP at 46 FPS. By combining the MobileNets framework with SSD we get more accurate real-time detection of objects which overcomes the drawback of traditional ML algorithms.

Figure 3: Objects Detected using YOLO Algorithm

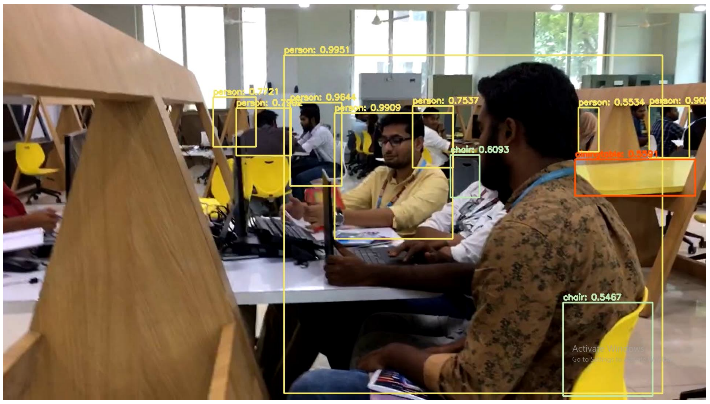

Figure 4: Persons and Objects detected in a live feed using Deep Learning based Algorithm

6. CONCLUSION

While we there is a demand for high speed data processing the machine learning techniques provide better scope for it. In this particular object detection we require the high available pre-trained datasets so that we can optimize error rate with inclusion of more parameters. In this work we have successfully identified the better technique for the real time object detection and in turn provided the drawbacks for other traditional algorithms. To further extent we have applied this approach to the home intrusion system wherein user gets notifications based on the logic given to system. Further to extend this work this system can be applied it to real time monitoring systems by providing the pre-defined inputs of our purpose and keeping a track of particular object.

REFERENCES

1. Ali Ibrahim , Member, IEEE, and Maurizio Valle, Senior Member, IEEE, **“Real-Time Embedded Machine Learning for Tensorial Tactile Data Processing”,** IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS–I: REGULAR PAPERS, VOL. 65, NO. 11, NOVEMBER 2018 https://doi.org/10.1109/TCSI.2018.2852260
2. Beatriz Blanco-Filgueira, Daniel Garcia-Lesta, Mauro Fernandez-Sanjurjo, Victor Manuel Brea and Paula L´opez, **“Deep Learning-Based Multiple Object Visual Tracking on Embedded System for IoT and Mobile Edge Computing Applications”**, 2327-4662 (c) 2018 IEEE.
3. Thiago Alves, Rishabh Das, and Thomas Morris, **“Embedding Encryption and Machine Learning Intrusion Prevention Systems on Programmable Logic Controllers”**, IEEE EMBEDDED SYSTEMS LETTERS, VOL. 10, NO. 3, SEPTEMBER 2018. https://doi.org/10.1109/LES.2018.2823906
4. Sheharyar Khan, Ahmad Farhan Latif and Sarmad Sohaib, **“Low-Cost Real-Time Non-Intrusive Appliance Identification and Controlling through Machine Learning Algorithm”**, 978-1-5386-4615-1/18/$31.00 _c 2018 IEEE
5. Jun Xiang, Guoshuai Zhang, And Jianhua Hou, **“Online Multi-Object Tracking Based on Feature Representation and Bayesian Filtering Within a Deep Learning Architecture”**, 10.1109/ACCESS.2019.2901520
6. Xiang Ying, Qiang Wang, Xuewei L, Mei Y, Han Jiang, Jie Gao, Zhiqiang Liu, And Ruiguo Yu, **“Multi-Attention Object Detection Model in Remote Sensing Images Based on Multi-Scale”**, 10.1109/ACCESS.2019.2928522
7. Yunhang Shen, Rongrong Ji , Senior Member, IEEE, Changhu Wang, Senior Member, IEEE, Xi Li, and Xuelong Li , Fellow, IEEE, **“Weakly Supervised Object Detection via Object-Specific Pixel Gradient”**, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, VOL. 29, NO. 12, DECEMBER 2018
8. Qichang Hu, Sakrapee Paisitkriangkrai, Chunhua Shen, Anton van den Hengel, and Fatih Porikli, **“Fast Detection of Multiple Objects in Traffic Scenes With a Common Detection Framework”**, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, VOL. 17, NO. 4, APRIL 2016
9. Hasith Karunasekera, Han Wang, **“Multiple Object Tracking With Attention to Appearance, Structure, Motion and Size”**, 10.1109/ACCESS.2019.2932301
10. Bernardo Augusto Godinho De Oliveira, Flávia Magalhães Freitas Ferreira And Carlos Augusto Paiva Da Silva Martins, **“Fast and Lightweight Object Detection Network: Detection and Recognition on Resource Constrained Devices”**, 10.1109/ACCESS.2018. 2801813
11. Sixian Chan, Xiaolong Zhou, Shengyong Chen, **“Robust Adaptive Fusion Tracking Based on Complex Cells and Keypoints”**, 10.1109/ACCESS.2017.2675438.
12. Maryam , Noor Akhmad Setiawan, **“A Wrapper Feature Selection Based on Ensemble Learning Algorithm for High Dimensional Data”**, International Journal of Advanced Trends in Computer Science and Engineering.
13. Mhalsakant Sardeshmukh, Vaishali Sardeshmukh, **“Performance Analysis of Human Detection and Tracking in Changing Illumination”**, International Journal of Advanced Trends in Computer Science and Engineering.
14. Nurul Amelina Nasharuddin, Nur Syamimie Mohd Yusoff, Siti Khadijah Ali, **“Multi-feature Vegetable Recognition using Machine Learning Approach on Leaf Images”**, International Journal of Advanced Trends in Computer Science and Engineering.