Designing and Implementing a Real time weapons detection system on ODROID-XU4

Seshadri Sastry Kunapuli
Scientist, AI
Xvidia Technologies, Gurugram
seshadri@xvidia.net

Praveen Chakravarthy BH
Research Engineer, AI
Xvidia Technologies, Gurugram
praveen@xvidia.net

Upasana Singh
Senior Research Engineer, AI
Xvidia Technologies, Gurugram
upasana@xvidia.net

Abstract*-Artificial Intelligence has got many applications in security and camera analytics. In recent times, there are 245 million IP cameras already installed globally. To make these IP Cameras intelligent, processing power is required. For which we may use a cloud or a local hardware which has pros and cons. But in general, selecting a local hardware to run an AI application is more difficult and almost a challenging task. According to latest statistics gun ownerships among civilians on a rise. Latest statistics say 393 million guns are owned by civilians in USA, 71 million guns are owned by civilians in India and 50 million guns are owned by civilians in China etc., which says threats due to guns are increasing globally. Existing weapon detection systems are costly and requires special equipment like special scanners, cameras etc. Weapon detection systems that are simple and accurate are required. Hence, in this paper, we proposed a design and implementation of weapon detection system in a simple ODROID-XU4 connected to a cheap IP HD camera and achieved an accuracy more than 99.8% in real time. So as to give an alert to the nearest police men if any weapon found on busy roads or in heavy buildings.
Keywords- Weapon Detection System, ODROID-XU4 CPU, IP Digital Camera

I. INTRODUCTION

Artificial Intelligence has a self-thinking capability in doing every day's, one's work in an intelligent way, example, Crowd Counting Algorithm to determine the traffic in a particular zone such as Railway stations, Bus Stations or inside any particular stores and Appearance Search Algorithm which is able to track a person from zone to zone by camera tweaks and face recognition which is able to do access controlling etc., and these processes can be done without using any Intelligent Camera(s) but making the cameras intelligent with the help of external devices connected to it. One such a device from many, (Beagle Boards and Raspberry Pi's) with energy-efficient hardware and small form factor is ODROID-XU4. To build such an intelligent system one has to know the architecture of the hardware so as to learn what kind of connections can be made on such a box. And also the second import task is that how to build an OS on such a hardware system and what kind of packets to be installed and how to be installed in such a hardware platform. For our convenience, we used SD Card instead of eMMC and switched the booting option to SD Card on the ODROID CPU. ODROID's can be used to build many robotic applications which is not ordinarily performed by either raspberry pi or Beagle boards so in this research paper, we are sharing our knowledge i.e., designing and implementing a real time weapon detection system on ODROID-XU4 with the help of a digital camera connected to a Wi-Fi module. For this project, we have used a 31.5 inch LED TV (of course not required), USB Keyboard and a mouse with a SanDisk Ultra HCI 16 GB SD Cards each one installed with different versions of Ubuntu/Linux Kernels. In one of the applications of the ODROID, a drone based positional estimation and fire detection system [1] using digital video colour space has been designed with the help of an ODROID-XU4 box.

In general, several applications like-wise (but less was conducted) can be designed and conducted by using an ODROID-XU4 box either with the hardware in designing Robots, Quad copters [1] etc., or by designing a light weight deep learning model for Computer Vision Applications. One such a process is done in this paper, i.e., designing and implementing a real time weapon detection system in an ODROID box. In general, many weapon detection systems have been proposed and only few of them are in the existence in the real time scenarios. Tuzi Xu et al [2], has proposed a concealed weapon detection system in which a hidden weapon was detected with the help of images obtained from multiple sensors are decomposed into low and high frequency bands by using double-density dual-tree complex wavelet transform (DDDTCWT). However this method seems to be deliver pretty results but cost and complexity of the system becomes a real burden in implementing at airports, bus stops or in any public transport areas. In the other research, Adaptive Neuro-Fuzzy Classifier for weapon detection in x-ray images of luggage using Zernike moments and shape context descriptor was proposed by Annet Deenu Lopez et al [3], which detects the shape of a concealed weapon in a luggage using X-ray machines as seen in the airports, with the help of the shape context descriptors. As we have

978-1-7281-1978-6/18/$31.00 ©2018 IEEE

seen in the airport's baggage counter there is an x-ray machine which scans the baggage for hidden weapons and electronic gadgets or batteries, the author takes those images and train them in a neural network for detecting the shape contexts of a weapon. But this paper is only helpful in finding the hidden weapons inside a baggage but it doesn't help in detecting the weapons concealed inside the coat of a person and also these kind of cameras are very costly to install it anywhere. There are many researches done in finding concealed weapon detection system [4, 5, 6] by using image sensors or by using image fusion techniques on infra-red images but most of them are not really implementable as some of them are very costly in installing and some of them are not usable for real time applications as in real time scenarios we need to consider many cases such as illumination, dust, climatic conditions and so on where the camera exists. So we trained a Deep learning model which can detect multiple weapons in the real time scenario and tested under different conditions in a clumsy area with different camera angles which can be seen in the results section V.

II. Designing a Real time weapon detection system

In this research, we designed a real time weapon detection system which detects multiple weapons such as AK-47, UZI, Air Pistol, Knives etc., even at a long range and wide camera angle(s). In this research, we have prepared our own weapon dataset from different Hollywood action movies nearly up to 3000 images and converted into a shape of 64*64, as the downloaded images could be of any resolution then we used the discrete wave let transform of the image and saved the LL Component from the images into another folder which is later split into trained and test folders before giving it to our DCNN model. After collected the images we have done one hot encoding to the labels of the different weapon folders in the dataset and have done image data augmentation technique with a scaling factor of 1.5 and later trained with our DCNN model. This architecture of the DCNN is so simple that it can be easily designed and implemented on a ODROID-XU4 system for a real time purposes even in a clumsy area which will be more helpful in railway stations, airports, bus stands etc., In this section we are going to explain the architecture and the concept behind our DCNN which is used to detect weapons from a camera feed.

The Architecture of our proposed weapon detection system consists of a Deep Convolution Network with two convolution layers having 5*5*3 filters which strides over the complete image and takes the dot product with the chunk of the image resulting 'k' feature maps whose value is equal to m-n+1. Where m = size of the image (either width/height), n is the size of the filter and the resulting value will be the k feature maps. In our network, as our image size is 64*64 and filter size is 5*5*3, the resulting feature maps generated will be 60 feature maps. So the size obtained from the output of first convolution layer will be (64*64, 60), i.e., 60 feature maps of size 64*64. Rectification linear unit (ReLu) activation function is used in the convolution layers so as to provide some non-linearity in the output function by keeping all the negative values obtained in the neurons to zeros. After convolution layer, we are taking the subsamples from all the 60 feature maps using maximum pooling function (size = 2*2), in order to reduce the spatial representation (i.e., from (64*64, 60) to (32*32, 60)), which alternatively reduce the computational parameters thus reducing the computational complexity of the network. We later on added one more convolution layer which defines the size of the new feature maps in the range of m+n-1 (whereas, m=32, n=5) and again max. Pooled to the shape (16*16, 28) as shown in the architecture below Fig. 3. The flatten layer now converts the three dimensional tensor into a mono dimensional tensor by converting (16*16, 28) into a (16*16*28) single dimension feature vector of size 7168 units.

Dense layers are the fully connected layers which was used in our network before our implied "softmax" classifier to determine the computational probabilities of our each class. The below figure depicts the architecture we have used to train our model in the ODROID-XU4 box in Fig.1. And our DCNN model architecture in the following Fig.2.

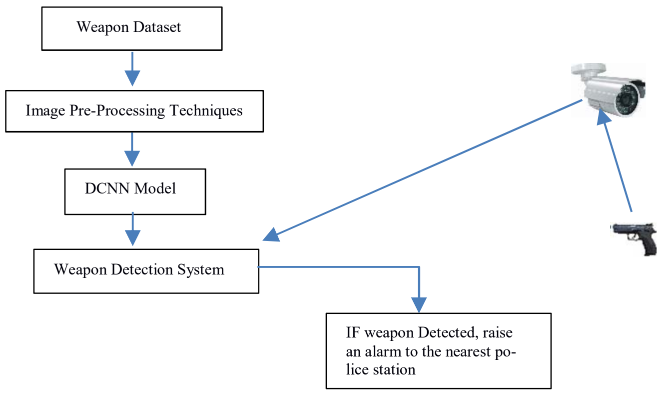

Fig.1. Architecture of our weapon detection system

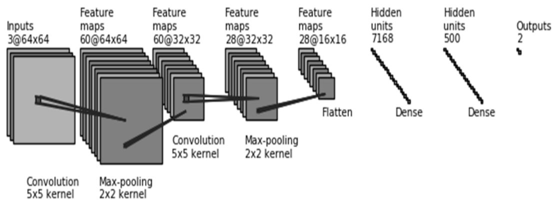

Fig. 2. DCNN Architecture of our weapon Detection System.

III. Proposed Architecture

For the easy understanding of our weapons detection system architecture, the functional flow diagram of our DCNN is shown in Fig.3 below, which consists of two convolution layers having a stride of 5*5 and two max. Pooling layers with a filter having pooling size of 2*2. A Flatten layer which converts these two dimensional vectors into one dimensional feature vector. And two dense layers which are implied before softmax layer classifier so as to predict the computational probabilities of the each class.

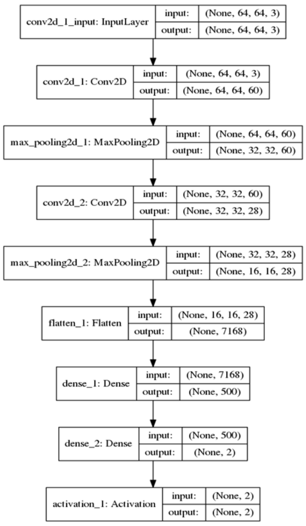

Fig. 3. Functional Flow diagram of our DCNN Architecture

The tensorflow architecture of the proposed method is shown in Fig. 4 below,

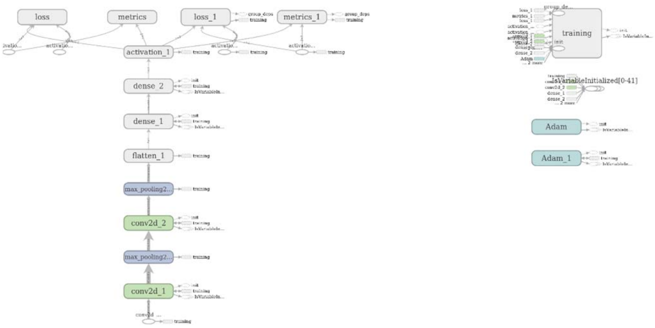

Fig. 4. Tensorflow Architecture of our proposed method

IV. Results

In this section, we are going to display the results of our DCNN model for real time weapon detection system using ODROID-XU4 CPU. As the model is a light weight model consisting of only two convolution layers (which is quite enough for our weapon detection system), this model takes only 39.8% of the total memory and 75% on aggregate of all cores of the cpu while training, which is as shown in the below Fig. 5

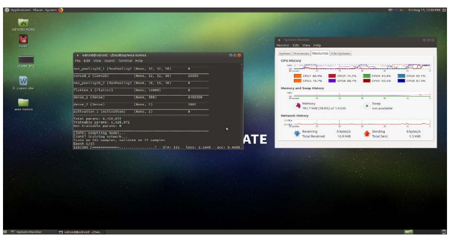

Fig. 5. CPU and Memory status while training the model using ODROID XU4 system

The below figures 6,7,9 depicts the outcome of our project which were taken on an IP camera in our office to detect our weapon detection system in a real time frame basis and figure 8 is a random picture chosen from internet which is famous raccoon. As you can see, if there is a weapon found in the frame then it will display the probabilities of the weapon found and non-weapon found in our frame which is the final output of the dense layer using softmax activation function. Note that, we are only classifying the frames containing weapon and non-weapon so as to ensure which frame the weapon contains (not the percentage of the weapon). This will be helpful to grab a frame in a densely crowded areas while monitoring a video and to alert the officials before anything can happen. And this architecture can detect merely 20 weapons as said earlier i.e., if any of those 20 categories found on the frame in a CCTV footage then it will alert the officials immediately without giving any delay. As you can see, we got merely 99% of accuracy even in a clumsy frame as shown in the below figures basis

Fig. 6. Frame displaying 99.98% accuracy that weapon is found in a frame which is grabbed from IP Camera (Note: the date and time are false in the frame) Source: Xvidia office cam

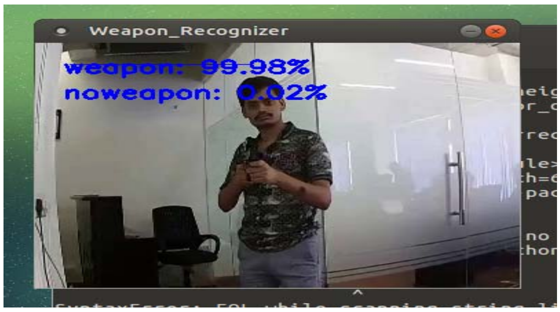

Fig. 7. One of the authors found carrying a weapon in his hand. Source: Xvidia office cam

Fig. 8. We have taken one image from the raccoon dataset just for fun, you can see that was giving 100% accuracy that, it was not a weapon (which means it was not considering the edges of the image as seen in some few existing models) Source: Internet

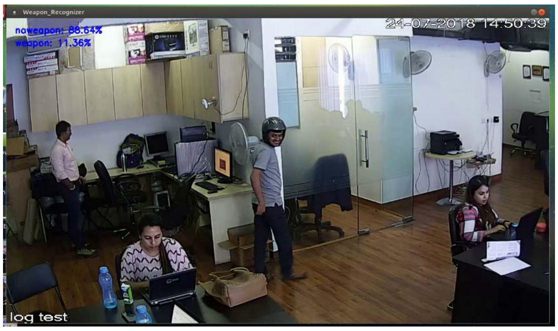

Fig. 9. Depicts a wide frame (50 degrees camera angle) covering a wide area of a clumsy room, so more edges but giving a higher probability for “weapon not found” in that frame. In the figure, you can see they are more edges and the cupboard above on one of the authors head (left top) having the shape of a weapon might give the display weapon value but the probability is less when compared to the “no weapon found” in the grabbed frame. (Source: Xvidia office cam)

Fig. 10 depicts the graph of training and validation accuracies and losses while training our model. As you can see, the training and validation losses are gradually decreasing while the accuracies of the training and validation increases for just 15 epochs.

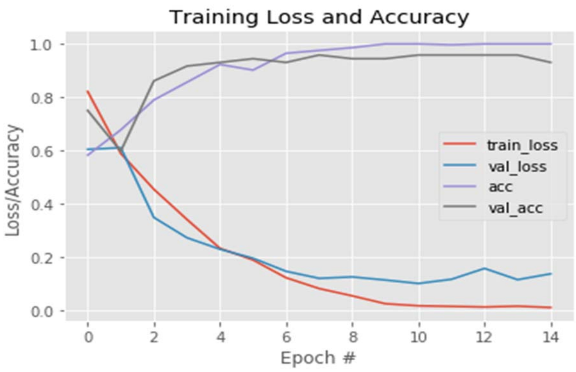

Fig. 10 Graph of Training Loss and accuracy for our model for only 15 epochs, as you can see the training accuracy became one and the validation accuracy was merely one. (i.e., 99.8% accuracy, as we are taking probabilities from max (0, 1))

The below table1 shows the evaluation report of our model in terms of Precision, Recall and F1-score for weapon and no weapon detection

Table1. Evaluation Report

Prediction	Precision	Recall	F1-Score	Support
no weapon	0.98	0.98	0.99	41
weapon	0.99	0.99	0.99	31
Avg/Total	0.99	0.99	0.99	72

V. Conclusion and Future scope

ODROID is a thunder packed on-chip single board computer with 2 GB Ram and 8 core processor. The main difference between the original CPU and the ODROID-XU4 box is the built in hardware kernels. With respect to the size, we can take it to everywhere by carrying it in your pocket. It will also be helpful in designing/implementing a real time model which is explained in this paper by building a weapon detection system with an accuracy of nearly 100%. One can use it in making the drones/robots for the real time purpose, which will be our future scope, detecting weapons through drones/mini-robots made up with Odroid box that will be helpful in army borders and highly delicate areas, where person can’t go. But the main drawback with this Odroid box is that it is quite costly when comparing to the raspberry pi kits but we can use this box in IFTTT also with the help of android operating system.

Acknowledgement- We sincerely thank Xvidia Technologies for providing us the cameras with different angles for testing purposes in the real time clumsy environments.

References

[1]. Muhammad Fadhil Abdullah, Inung Wijayanto, Angga Rusdinar, Position Estimation and Fire Detection Based on Digital Video Color Space for Autonomous Quad copter Using Odroid XU4, 016 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC)

[2]. Tuzhi Xu ; Q. M. Jonathan Wu, Multisensor concealed weapon detection using the image fusion approach, 6th International Conference on Imaging for Crime Prevention and Detection (ICDP-15)

[3]. Annet Deenu Lopez ; Eldho S. Kollialil ; K. Gopika Gopan, Adaptive Neuro-fuzzy Classifier for Weapon Detection in X-Ray Images of Luggage Using Zernike Moments and Shape Context Descriptor, 2013 Third International Conference on Advances in Computing and Communications

[4]. Z. Xue and R.S. Blum, “Concealed weapon detection using color image fusion,” in Proc. Int. Conf. on Image Fusion, Queensland, Australia, 2003, pp. 622-627.

[5]. H. Chen, S. Lee, R.M. Rao, M. Slamani, and P.K. Varshney, “Imaging for concealed weapon detection: a tutorial overview of development in imaging sensors and processing,” IEEE Signal Process. Mag. , vol. 22, no. 2, pp. 52-61, Mar. 2005

[6]. Y. Wang and M. Lu, “Image fusion based concealed weapon detection,” in Proc. Int. Conf. on Computational Intelligence and Software Engineering , 2009, pp. 1-4.

[7]. B.C.K. Tso and P.M. Mather, “Classification of multisource remote sensing imagery using a genetic algorithm and Markov random fields”, IEEE Geoscience and Remote Sensing Society, pp. 1255 – 1260.

[8]. Stelios K. Mylonas, Dimitris G. Stavrakoudis, John B, Theocharis and Paris A. Mastorocostas, “Classification of Remotely Sensed Images Using the GeneSIS Fuzzy Segmentation Algorithm”, IEEE Geoscience and Remote Sensing Society, pp. 5352 – 5376

[9]. Daijin Kim and Sung-Yang Bang, “A handwritten numeral character classification using tolerant rough set”, IEEE Computer Society, pp. 923 – 937

[10]. Zhongwei Sun, Keyong Hu, Tong Hu, Jing Liu and Kai Zhu, “Fast Multi-Label Low-Rank Linearized SVM Classification Algorithm Based on Approximate Extreme Points”, IEEE, pp. 42319 – 42326

[11]. Lipo Wang, Feng Chu and Wei Xie, “Accurate Cancer Classification Using Expressions of Very Few Genes”, IEEE/ACM Transactions on Computational Biology and Bioinformatics, pp. 40 – 53