HARDWARE IMPLEMENTATION OF MULTI OBJECT RECOGNITION SYSTEM USING MOVIDIUS NEURAL NETWORK

A. Moncy Amala Infanta

Department of Electronics and Communication Engineering, PG Scholar, National Engineering College, Kovilpatti, Tamil Nadu, India

T. Devakumar

Department of Electronics and Communication Engineering, Assistant professor(SG), National Engineering College, Kovilpatti, Tamil Nadu, India

---***---

Abstract- Vision is the key component in Artificial Intelligence and automated robotics. Detection of Multiobject in the Real time image is one of the main expected functionality to be executed in Real Time video surveillance. This functionality is executed by an Intelligent System attached with that Camera. This project deals with the design of that intelligent system using the concept of Artificial Intelligence. Thus the prediction of Multiobject in the present captured image with the previous knowledge is done by that designed Intelligent System. In order to detect the Multiobject in the present capture image, a computing board is used, which compares the captured image with a dataset and identifies the various stages of the object. In this paper we used high speed graphics processors and identified which one has more accuracy.

Keywords- Next Unit Computing Board, MOVIDIUS NEURAL Stick, Multiobject Detection.

1. INTRODUCTION

Object detection and detection of human Identification is one of important in video surveillance. In this project to detect the various objects in the single frame by using YOLO techniques. In Image processing dataset creation is the main thing get good quality images with good resolutions. Then Image preprocessing is elimination of noises in the image by using binning method or clustering or kernel or regression or moving average or curve fitting methods. Then Image segmentation is identifying the object region. In feature extraction is to extract the main object region in matrix values based on the color or ridges. There are many extract methods like shape or texture or wavelet or Laplacian of Gaussian or first order intensity for statistical methods. In this paper YOLO is used for image classification by applying machine learning algorithms. Lastly detection of Object in inference stage using learners is supervised and unsupervised learning. Nowadays various improvements in the field of agricultural robotics, an automated robot is developed and has to climb the coconut trees and detect the objects, though execute this functionality on an intelligent system attached with that machine. In coconut harvesting machines, Camera plays a vital role. After climbing the tree, a connected camera captures the image compared with a trained dataset and detect the object. In the stage of identification the dataset must be trained. Use a computing board board to train and deploy on those dataset. In the training stage YOLO is used and it trains six frames in a single second.

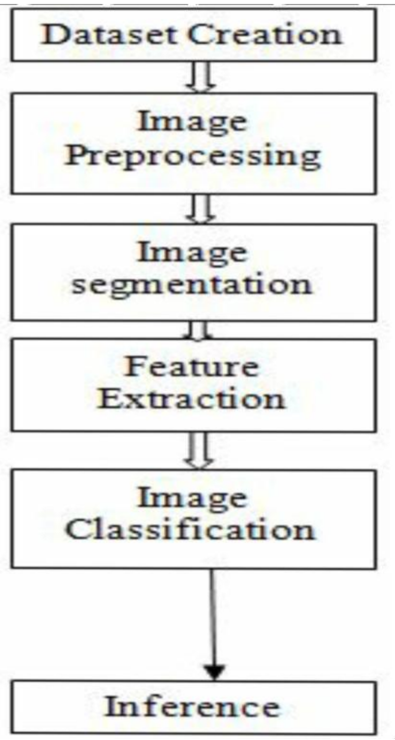

Figure 1. Basic Architecture of image processing

2. RELATED WORKS

To Design the Intelligent system to provide required data and trained that machine has able to detect the object. This system has improved by a lot of literature that various types of objects. There are many literatures that use contour identification, canny edge detector, laplacian of Gaussian, PSO, technique to detect the shapes and colors of the objects. Contour based identification technique identifies the edge of the object. Canny edge and laplacian of Gaussian technique has identified the outlines of the objects.

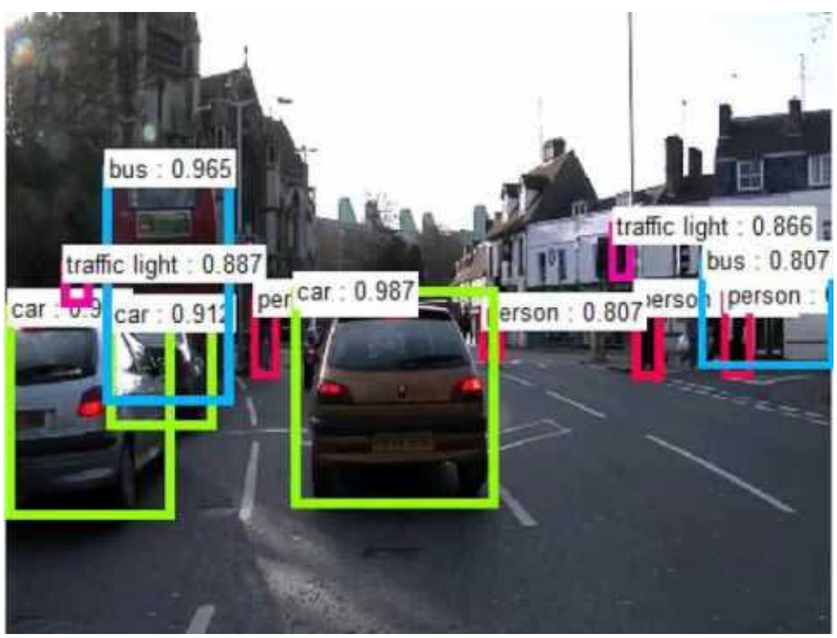

Figure 2. Multiobject Detection

Particle Swarm Optimization is one of the methods to find the best position of the objectt based on the image captured input image. But its drawback is difficult to find the starting position of the co-ordinates. All this previous work has more validation loss in the images.

3. METHODOLOGY

In this paper two MOVIDIUS NEURAL Stick has used for training yolo network is used. I collected 300 sample images to segment the correct object by using Background Technique. And train the images in YOLO (You Only Look Once).

A. YOLO Workflow

- **Pre-train** a CNN network on image classification task.
- **Split** an image into S×S cells. If an object's center falls into a cell, that cell is "responsible" for detecting the existence of that object. Each cell predicts (a) the location of B bounding boxes, a confidence score, and a probability of object class conditioned on the existence of an object in the bounding box.
- The coordinates of bounding box are defined by a tuple of 4 values, (center x-coord, center y-coord, width, height) where xx and yy are set to be offset of a cell location. Moreover, x, y, w and h are normalized by the image width and height, and thus all between (0, 1)

B. Network Architecture

The base model is similar to googlenet with inception module replaced by 1x1 and 3x3 convolution layers.

PROPOSED METHODOLOGY

Computing (NUC) Board under Ubuntu linux os is used to execute the SSD method. A Logitech 720p camera has connected with NUC Board capture the image and detection of objects. The NUC Board process, the captured image has preprocessed by using morphological operation. By using SSD technique, the collected images has trained well, after the training a (.h5) file has generated. This file has predicted the object with real time images. After the trained stage we get the trained model, by using a USB camera to capture the input image and compare the trained model and detect the coconut.

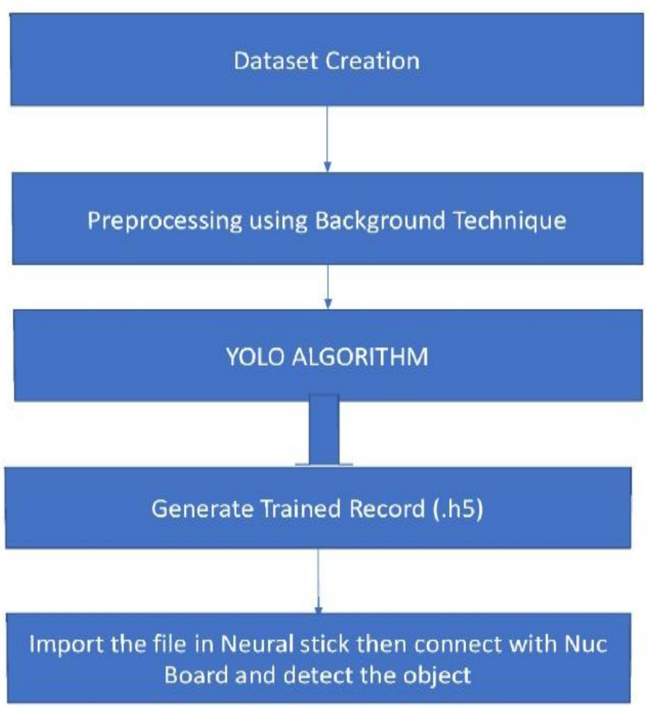

Figure 3. Proposed Methodology

BACKGROUND TECHNIQUE

In Background Technique has filtered the background of the and detect the shape of the object. It has filtered in two types are recursive and non recursive. In this paper Recursive method has used. It has filtered and detect the shapes and gives good accuracy result. YOLO is used to train the network. After training (.h5) file is generated this file has import to Intel MOVIDIUS NEURAL Stick. To connect the stick with Nuc board and detect the Multiobject in single frame.

Dataset Creation

I collected various images like human images, monitor, birds image etc. To preprocessed with background technique. Train all images in YOLO network.

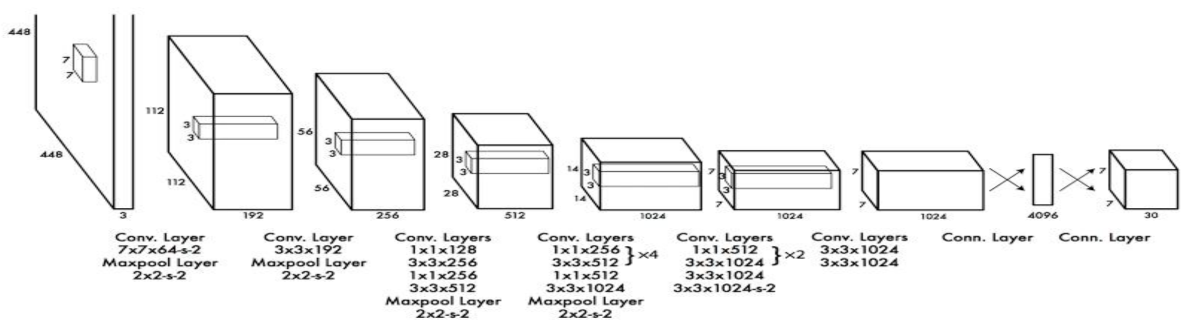

Figure 1. YOLO OPERATION

Loss Function

The loss consists of two parts, the *localization loss* for bounding box offset prediction and the *classification loss* for conditional class probabilities. Both parts are computed as the sum of squared errors. Two scale parameters are used to control how much we want to increase the loss from bounding box coordinate predictions (λcoord) and how much we want to decrease the loss of confidence score predictions for boxes without objects (λnoobj). Down-weighting the loss contributed by background boxes is important as most of the bounding boxes involve no instance. In the paper, the model sets λcoord=5 and λnoobj=0.5.

Hardware Requirements

INTEL MOVIDIUS NEURAL STICK
VPUs drive the demanding workloads of modern computer vision and AI applications at ultra-low power. MOVIDIUS technology

Figure 2. MOVIDIUS NEURAL STICK

allows device makers to deploy deep neural network and computer vision applications in categories such as smartphones, drones, intelligent cameras and augmented reality devices.
The intel Movidius Neural Compute Stick (NCS) is a tiny fanless deep-learning device that can be used to learn AI programming at the edge. NCS is powered by the same low-power, high-performance Intel Movidius Vision Processing Unit that can be found in millions of smart security cameras, gesture-controlled drones, industrial machine vision equipment, and more. Supported frameworks are TensorFlow and Caffe

Next Unit Computing Board

Computing Board (NUC) is a line of small-form-factor bare-bone computer kits designed by Intel. The NUC has had eight generations so far, spanning from Sandy Bridge-based Celeron CPUs in the first generation through Ivy Bridge-based Core i3 and i5 CPUs in the second generation to Gemini Lake-based Pentium and Celeron CPUs and Kaby Lake-based Core i3, i5, and i7 CPUs in the seventh and eighth generations. The NUC motherboard measures 4 × 4 inches (10.16 × 10.16 cm).

The bare-bone kits consist of the board, in a plastic case with a fan, an external power supply, and a VESA mounting plate. Intel does sell just the NUC motherboards, which have a built-in CPU, although (as of 2013) the price of a NUC motherboard is very close to the corresponding cased kit third-party cases for the NUC boards are also available.

Processor

- A soldered-down 7th generation Intel® Core™ i5-7260U dual-core processor with up to a maximum 15 W TDP

1. Intel® Iris™ Plus Graphics 640.
2. Integrated memory controller.
3. Integrated Platform Controller Hub.

BIOS –Basic Input/ Output System

- Intel® BIOS resident in the Serial Peripheral Interface (SPI) Flash device
- Support for Advanced Configuration and Power Interface (ACPI), Plug and Play, and System Management BIOS (SMBIOS)

Peripherals

USB3.0, USB2.0

Software Requirements

Python3, opencv

4. RESULT AND DISCUSSION

In this work Computing Board under Ubundu linux os is used to conduct the image processing with YOLO method. USB camera as input to capture the image is also connected with NUC Board. In order to start NUC Board process, the raw captured image has been filtered by using YOLO technique and more accurate for determining the solution. The filter consists of blurring, coloring, brightness, contrast and HSV (Hue, Saturation, Value) filter. It helps to reduce the noise from the condition around the coconuts at the tree, such as leaves, trunk, dry wood, sunlight and etc.

Train loss	Train accuracy	Validati -on loss	validation Accuracy
1.4349	0.6155	1.21	0.7479
0.7432	0.76	1.433	0.7863
0.7951	0.7791	1.0	0.7949
0.9577	0.7255	1.67	0.7137
0.690	0.8056	1.94	0.6923
0.6349	0.8043	1.67	0.6581
0.4497	0.8660	0.598	0.8462
0.5148	0.8531	0.534	0.8846
0.4820	0.8367	1.184	0.7479
0.3311	0.9009	0.552	0.8718

Figure 3. Training period Table

Figure 4. Training Loss and Accuracy

5. CONCLUSION

Due to its powerful learning ability and advantages in dealing with occlusion, scale transformation and background switches, deep learning based object detection has been a research hotspot in recent years. This paper provides a detailed review on deep learning based object detection frameworks which handle different sub-problems, such as occlusion, clutter and low resolution, with different degrees of modifications on R-CNN. The review starts on generic object detection pipelines which provide base architectures for other related tasks. Then, three other common tasks, namely salient object detection, face detection and pedestrian detection, are also briefly reviewed. Finally, we propose several promising future directions to gain a thorough understanding of the object detection landscape. This review is also meaningful for the developments in neural networks and related learning systems, which provides valuable insights and guidelines for future progress.

6. REFERENCES

[1] P. F. Felzenszwalb, R. B. Girshick, D. Mcallester, and D. Ramanan, "Object detection with discriminatively trained part-based models," IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 9, p. 1627, 2010.

[2] K. K. Sung and T. Poggio, "Example-based learning for view-based human face detection," IEEE Trans. Pattern Anal. Mach. Intell., vol. 20, no. 1, pp. 39–51, 2002.

[3] C. Wojek, P. Dollar, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," IEEE Trans. Pattern Anal. Mach. IntellIntell., vol. 34, no. 4, p. 743, 2012.

[4] H. Kobatake and Y. Yoshinaga, "Detection of spicules on mammogram based on skeleton analysis." IEEE Trans. Med. Imag., vol. 15, no. 3, pp. 235–245, 1996.

[5] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in ACM MM, 2014.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in NIPS, 2012.

[7] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in CVPR, 2017.

[8] Z. Yang and R. Nevatia, "A multi-scale cascade fully convolutional network face detector," in ICPR, 2016.

[9] C. Chen, A. Seff, A. L. Kornhauser, and J. Xiao, "Deepdriving: Learning affordance for direct perception in autonomous driving," in ICCV, 2015.

[10] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," in CVPR, 2017.

[11] A. Dundar, J. Jin, B. Martini, and E. Culurciello, "Embedded streaming deep neural networks accelerator with applications," IEEE Trans. NeuralNeural Netw. & Learning Syst., vol. 28, no. 7, pp. 1572–1583, 2017