PR1101: Automation Project (IoT)

Institute of Engineering & Technology (IET)

JK Lakshmipat University, Jaipur

Project Report

OBJECT DETECTION USING JETSON NANO

Submitted by:

Akshi Agarwal: 2019/btech/cse003

Ishika Gupta: 2019/btech/cse023

Prachika Kanodia: 2019/btech/cse068

Submitted to:

Dr.Devika Kataria

Mr. Divanshu Jain

Mr. HP Agarwal

Prof. Gustavo Sanchez

CERTIFICATE

This is to certify that the Automation Project work entitled "**Object Detection having text capability using Nano Jetson Board"** submitted by **Prachika Kanodia (2019BTechCSE068), Ishika Gupta (2019BTechCSE023), Akshi Agarwal (2019BTechCSE003).** In our opinion, the submitted work has reached a level required for being accepted for Automation Project.

Dr. Devika Kataria	Prof. Gustavo Sanchez
Name of the faculty supervisor	Name of the faculty supervisor
Department of Electrical and Electronics Engineering	Department of Electrical and Electronics Engineering
Institute of Engineering & Technology	Institute of Engineering & Technology
JK Lakshmipat University Jaipur	JK Lakshmipat University Jaipur

Mr. Divanshu Jain	Mr. H.P. Agarwal
Name of the faculty supervisor	Name of the faculty supervisor
Department of Electrical and Electronics Engineering	Department of Electrical and Electronics Engineering
Institute of Engineering & Technology	Institute of Engineering & Technology
JK Lakshmipat University Jaipur	JK Lakshmipat University Jaipur

CONTRIBUTION

The work has been done and equally managed by all the group members.

- Research
- Board Connections
- Coding
- Report
- AI Applications

Akshi Agarwal: 2019/btech/cse003

Ishika Agarwal: 2019/btech/cse023

Prachika Kanodia: 2019/btech/cse068

ABSTRACT

Jetson Nano is a GPU-enabled edge computing platform for AI and deep learning applications. The GPU-powered platform is capable of training models and deploying online learning models but is most suited for deploying pre-trained AI models for real-time high-performance inference. Using additional Nvidia tools, deployed standard AI models can have enhanced fidelity performance. The main objective of our project is to detect objects in real life with the help of jetson nano. We were able to achieve this objective with the help of SD-MobileNet-v2 model, which stands for single-shot detection on mobile devices. The detection is quite accurate but it was noticed that it fails to distinguish between a few objects.

LIST OF TABLES:

LIST OF FIGURES:

CHAPTER 1: INTRODUCTION

OBJECT DETECTION:

Object detection is the technique of determining the presence of an object and estimating its location in the image canvas. Object recognition classifies the detected object from the list of previously seen (trained on) objects. Detection techniques usually form a rectangular bounding box around the object and is a coarse representation of the extent of the object in the image. Recognition techniques usually classify the object with a certain probability or belief in the prediction.

In an image with multiple objects, it is a challenging task to determine the location of all the individual objects (detection) and then recognize them, due to several reasons:

- There can be a possible overlap between multiple objects causing occlusions for one or all.
- Objects in the image can have varying orientations.
- The objects could only be partially present in the image.
- Images from low fps video stream can be blurry and distort the features of the object.

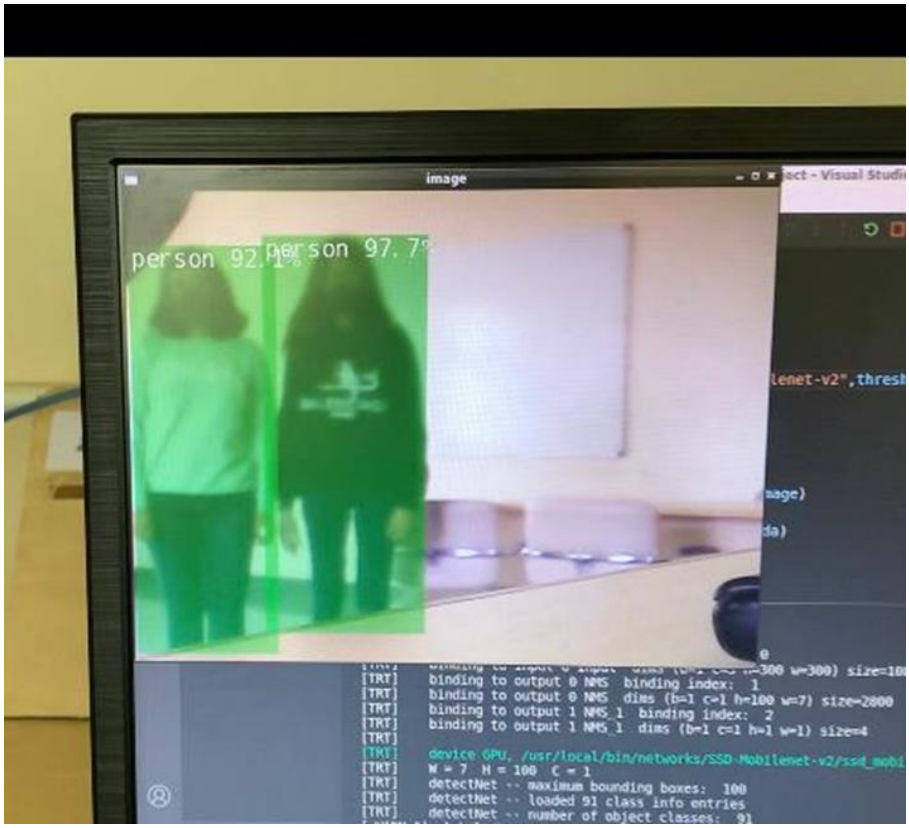

Fig 1. Object detection with jetson nano board

Among the provided models, we use the SSD-MobileNet-v2 model, which stands for single-shot detection on mobile devices. It is a part of the DetectNet family. This model is pre-trained on the MS COCO image dataset over 91 different classes. It can detect multiple objects in the same frame with occlusions, varied orientations, and other unique nuances. The model is pre-trained on common objects like soda cans, ovens, toasters, TVs, cakes, pizzas, and several other everyday items.

CHAPTER 2: PROJECT OBJECTIVE AND STATEMENT

To detect and recognize the object using SSD-MobileNet V2 model.

- We want to create a model for visually impaired people that will assist them in learning about their surroundings by telling them the name of the object. As our model detects and recognizes the object, it will speak its name aloud to learn more about the object in front of them.

CHAPTER 3: LITERATURE REVIEW

According to author, Drone object detection is a critical location of studies because it has many packages in real-time. In this evaluate paper, present studies works are studied. The works are labelled in keeping with their packages and techniques. This paper explores deep mastering and conventional photo processing techniques used for drone item detection. Dataset and assessment metrics also are discussed. The literature exhibits that deep mastering algorithms carry out higher than conventional photo processing techniques. Although the drones are small in length and regulations in power, it's miles critical to layout an green deep mastering set of rules for drone item detection. Variations perspective is some other giant mission to be addressed via way of means of the item detection algorithms. As a destiny work, we aimed to research item detection the use of UAV in agriculture application. Precision agriculture offers to reveal the fields exactly via way of means of amassing and reading the data. Acquiring aerial pictures the use of UAV are much less pricey than the satellites. Object detection in UAV aerial pictures in precision farming is an critical studies location to be investigated. We proposed a steady onboard item detection framework in precision agriculture on this paper and enforcing it'll be our destiny work.

With the increase in the availability of the internet, people are consuming more and more content on a regular basis and hence the chances of getting exposed to explicit content increase exponentially. To detect this explicit content from images or videos we make use of Convolutional Neural Networks (CNNs). This paper highlights the trade-off between speed, accuracy and training methodology for explicit content detection using Faster R-CNN and SSD MobileNet v2. FRCNNs introduce Regional Proposal Networks (RPNs) replacing the Search selective process thus making it faster for object detection. On the other hand, the combination of the MobileNet v2 architecture and the Single Shot Detector(SSD) framework yield an efficient object detection model making use of depth wise separable convolution. This paper highlights the difference in performance between two models object detection models. Taking advantage of the considerably smaller size of MobileNet v2, this model performed better as compared to Faster R-CNN in terms of speed especially on videos yielding more Frames per second on our test machine. Hence, MobileNet v2 can be used in real time object detection. However, with more training data, both models can perform considerably better.

The author examined several object detection, tracking, and recognition approaches, feature descriptors, and segmentation methods based on video frames and various tracking technologies. With new concepts, this approach was applied to improve object detection. Furthermore, tracking

the object from video frames is supplied in the bibliography section, along with a theoretical explanation. Because it will lead to a new subject of research, the bibliography content is the most significant contribution of research. We recognised and evaluated the limitations and future potential of various strategies. In addition, we identified certain approaches that provide accuracy but have a high computational complexity. Statistical approaches, background subtraction, and temporal differencing with optical flow were specifically studied. However, these techniques must focus on dealing with abrupt lighting changes, heavier shadows, and object occlusions.

CHAPTER 4: SCHEMATIC DIAGRAM

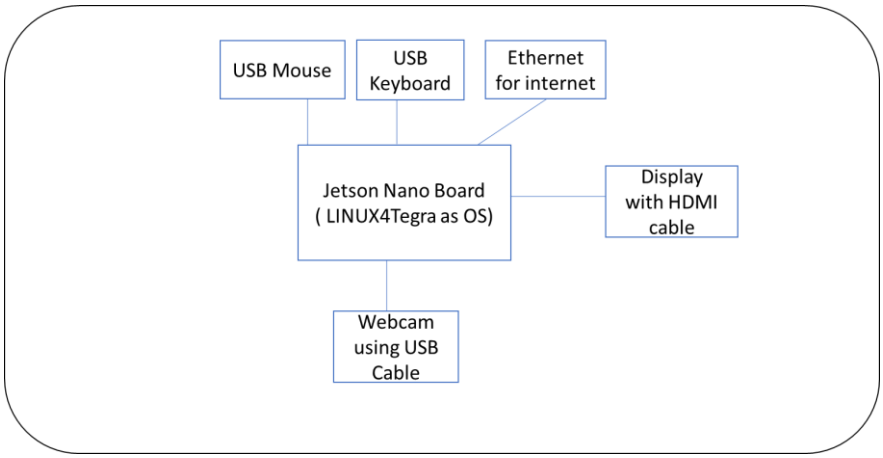

Fig 2. Schematic Diagram

CHAPTER 5: COMPONENTS USED AND COST

Table 5.1 Components Used and Cost

PRODUCT USED	COST (Rs.)
NVIDIA Jetson Nano Developer Kit	11,200
Memory Card 128GB	1,200
Logitech Webcam	829
C- type Cable	115
USB Mouse	399
Ethernet Cable	150
USB Keyboard	699
HDMI Cable	300
TOTAL:	14,892

CHAPTER 6: SOFTWARE AND LIBRARIES INSTALLED

SOFTWARES:

1. **Linux4Tegra:**

Linux for Tegra (Linux4Tegra, L4T) is a Linux based system software distribution by Nvidia for the Tegra processor series, used in platforms like the Nvidia Jetson board series.

This system software comes with JetPack, a Software Development Kit (SDK) from Nvidia. The official Nvidia download page bears an entry for JetPack 3.2 (uploaded there on 2018-03-08) that states:

JetPack 3.2 adds support for the Linux for Tegra r28.2 image for the Jetson OS. It is packaged with newer versions of Tegra System Profiler, TensorRT, and cuDNN from the last release.

2. **VS Code:**

Visual Studio Code is a source-code editor made by Microsoft for Windows, Linux and macOS.[9] Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

LIBRARIES:

1. **OpenCv2(4.5.4v):**

OpenCV is a great tool for image processing and performing computer vision tasks. It is an open-source library that can be used to perform tasks like face detection, objection tracking, landmark detection, and much more. It supports multiple languages including python, java C++.

2. **Jetpack SDK(4.6v):**

NVIDIA JetPack SDK is the most comprehensive solution for building AI applications. It bundles all the Jetson platform software, including TensorRT, cuDNN, CUDA Toolkit, VisionWorks, Streamer, and OpenCV, all built on top of L4T with LTS Linux kernel.

3. **Model:**

MobileNet is an object detector released in 2017 as an efficient CNN architecture designed for mobile and embedded vision application. This architecture uses proven depth-wise separable convolutions to build lightweight deep neural networks.

The basic parameters involved in it are:

- Depthwise Separable Convolution
- Network Structure and Training
- Width Multiplier: Thinner Models
- Resolution Multiplier

The steps involved in implementation of MobileNetV2 on video streams are as follows:

1. Install the TensorFlow Object detection API.
2. Download the model file from the TensorFlow model zoo.
    - Download the MobileNetV2 pre-trained model to your machine
    - Move it to the object detection folder.
    - Create a python script to run the real-time program.
    - It helps us load images or convert real-time video frames into NumPy arrays.
3. Setting up the configuration file and model pipeline.
    - Identifying the path to the pipeline config of our MobileNetV2 model. This configuration file defines the model architecture and params.
    - Specifying the checkpoint file of the model to be used.
4. Initialize model prediction by passing in the config path of the model.
    - Use TensorFlow to restore the model's last checkpoint by specifying the checkpoint directory.
    - Pre-process the image.
    - Assign a target label to the object in the image.
    - Predicts the probability of the target label to each frame in the image.
5. Create a script to put them together.
    - Now that we have the video stream and the writer in place, the next step is to keep the video stream live and perform real-time object detection by looping through the frames catpured from the video stream. As long as this keeps running, we can visually see the object detection result by displaying it on our screen.

Finally, once the stream goes off, the video writer then converts all frames captured so far into a video (with the real-time object detection result). The below code helps us get this done from end-to-end.

CHAPTER 7: RESULT AND DISCUSSION

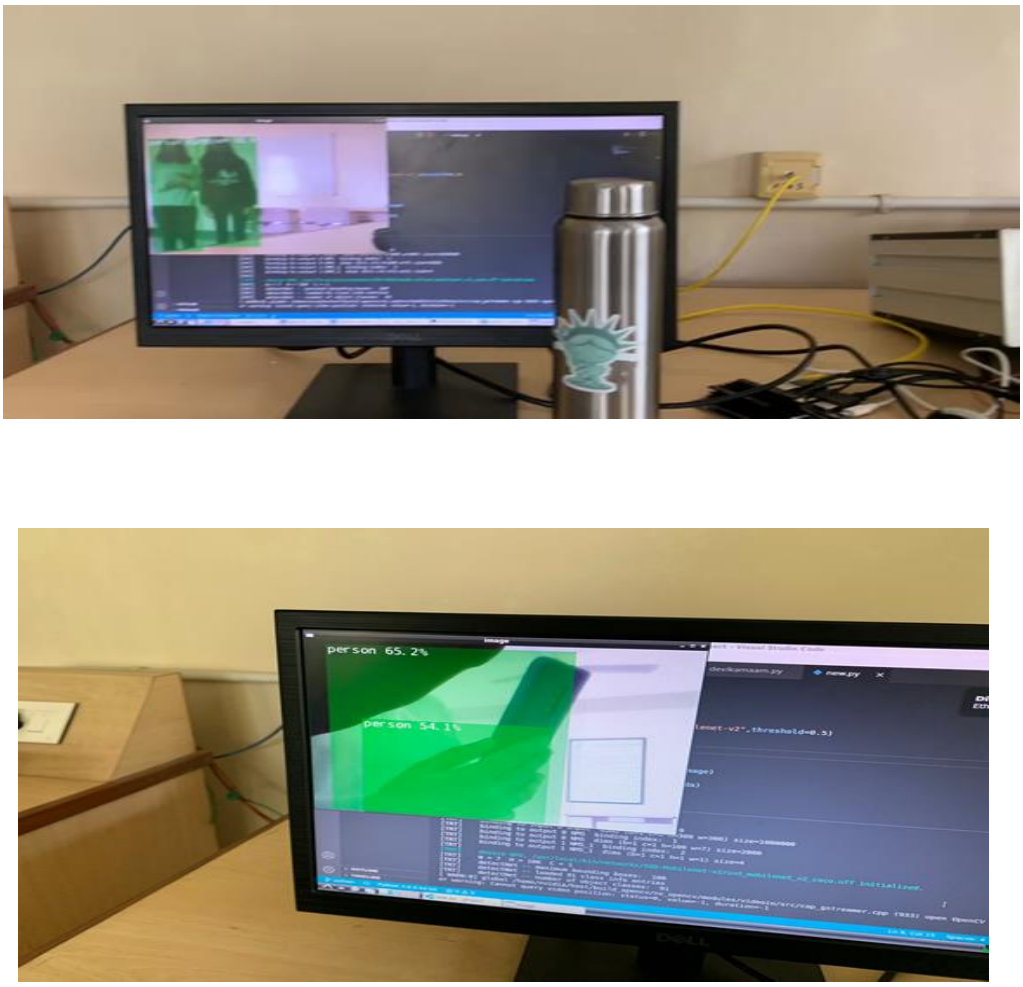

Fig 3. Object detection

We used the MobileNetSSD2 model for object detection and recognition. The precision of this model is good, although it fails to distinguish between a few objects.

FUTURE SCOPE:

1. In future we decide to add speech ability when object is detected along with text.
2. Also we will try to train the model with more epochs and accuracy.

CHAPTER 8: LEARNING OUTCOMES

1. We have learned about the structure of NANO Jetson Board and its working to implement the AI applications.

2. Also we learned that how the AI applications are implemented using this board i.e., object detection having text capability.

CHAPTER 9: REFERENCES

- https://www.irjet.net/archives/V6/i9/IRJET-V6I9144.pdf
- Image Processing. (2020, February 17). Medium. Retrieved December 23, 2021, from https://hicraigchen.medium.com/digital-image-processing-using-fourier-transform-in-python-bcb49424fd82
- nvidia jetson. (2019, May 6). Pyimagesearch. Retrieved December 23, 2021, from https://www.pyimagesearch.com/2019/05/06/getting-started-with-the-nvidia-jetson-nano/

CHAPTER 10: CODE

```
import cv2
#from cv2 import displayOverlay
import jetson.inference
import jetson.utils


net = jetson.inference.detectNet("ssd-mobilenet-v2",threshold=0.5)
cam= cv2.VideoCapture(0)
cam.set(3,640)
cam.set(4,480)
while True:
    sucess,image=cam.read()
    imgCuda = jetson.utils.cudaFromNumpy(image)
    detections = net.Detect(imgCuda)
    image = jetson.utils.cudaToNumpy(imgCuda)
    cv2.imshow("image",image)
    cv2.waitKey(1)
```