ISSN: 2454-9940

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

E-Mail :
editor.ijasem@gmail.com
editor@ijasem.org

www.ijasem.org

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN 2454-9940

www.ijasem.org

Vol 18, Issue 4, 2024

AN INTERNET OF THING BASED OBJECT CLASSIFICATION USING EDGE IMPULSE AND RASPBERRY PI PLATFORM WITH DIGITAL HUMIDITY AND TEMPERATURE SENSOR

Zainab Unnisa

Master of Engineering, Nawab Shah Alam Khan College of Engineering & Technology

Dr. Shanila Mahreen

HoD, Associate professor,

Nawab Shah Alam Khan College of Engineering & Technology

***Abstract*—** Object Detection is widely utilized in several applications such as detecting vehicles, face detection, autonomous vehicles and pedestrians on streets. TensorFlow's Object Detection API is a powerful tool that can quickly enable anyone to build and deploy powerful image recognition software. Object detection not solely includes classifying and recognizing objects in an image however additionally localizes those objects and attracts bounding boxesaround them. This paper mostly focuses on setting up the environment and tflite model for detecting hotel items like jug, cup and flask. We have got Tensor flow Object Detection API to train model and we have used Single Shot Multibox Detector (SSD) MobileNet V2 algorithm for implementation.

Key words: TensorFlow, Transfer Learning, Edge Impulse, Neural network: Convolutional Neural Network, SSD-Mobilenet-V2.

1. *Introduction*

The study of comprehending and modifying digital images and videos is known as computer vision (CV). Numerous applications, such as face recognition, image retrieval, industrial inspection, augmented reality, and more, depend heavily on computer vision. Deep learning has made computer vision effective for many applications. A subfield of machine learning called deep learning uses a group of techniques called artificial neural networks, or ANNs. Artificial neural networks (ANNs) are modeled on the human brain, with nodes that are connected to one another and exchange data. There are several categories into which deep learning for computer vision can be applied: video and picture production, segmentation, detection, and classification. The entire image is labeled by the

ISSN 2454-9940
www.ijasem.org
Vol 18, Issue 4, 2024

image classification. Object localization is the process of identifying an object's location in addition to labeling it. Rectangular coordinates usually define the object's position. Detection is the process of using rectangular coordinates to find many items in an image. The process of segmentation involves identifying precise objects by overlaying them with a transparent mask that has precise edges. All an image classification or image recognition model does is identify the probability of an object in an image. In contrast, an object's position in the image corresponds to its location. A localization procedure for an object will provide the item's location coordinates in relation to the image or picture. Object detection is a significant CV problem. Similar to image classification challenges, detection performance has been improved via deeper networks. The accuracy of these methods is currently very good. It is employed in numerous applications as a result. The quantity of items is the difference. There are a variety of objects in detection. When developing the architectures for the deep learning model in terms of detection or localization, this minor distinction has a significant impact.

2. *Literature Survey*

- "Gayathri *et al* (2019) has proposed a faster model of object detection namely Single Shot Multi-Box Detection (SSD) along with Mobile Net, published in second International Conference on Computational Intelligence in Data Science (ICCIDS-2019)."

"It is shown that for real time applications where inference needs to be quick, this model does the work elegantly. The paper compares other deep learning models like RCNN, Fast RCNN, Faster RCNN, but owing to their complex networked structure, the accuracy is lesser when detecting multiple objects in a single frame. As SSD uses convolution filters for detecting the objects depth it loses its accuracy if the frame is of low resolution. So we use the Mobile Net technique as it use depth wise separable convolution which significantly reduces the parameters size when compare with normal convolution filters of same depth. Thus, we can get the light weight deep neural network as a result. On whole SSD with Mobile Net is nothing but a model in which Meta architecture is SSD and the feature extraction type is Mobile Net."

ISSN 2454-9940
www.ijasem.org
Vol 18, Issue 4, 2024

- "Abdelhakim Zeroual (2019) *et al* published in the Proceedings of the International Conference on Networking and Advanced Systems (ICNAS) IEEE and proposed that there are many deep neural network frameworks for embedded platforms. Caffe2 is a framework, which cares multiple embedded platforms such as, Android, iOS, Tegra, and Raspbian. TensorFlow Lite is another is a lightweight and accessible framework and a new trend, which supports a variety of embedded platforms. We can implement and run deep neural network using Python and C++. In the domain of security on Mobile Cloud Environment (MCC), for facial recognition based on deep convolutional neural network to authenticate the device, the training and recognition phases were done on cloud because of huge computation. Based on this work, we propose to use the TensorFlow lite for mobile device to reduce the time of transfer between cloud and mobile. By focusing on the cited work, the authors have achieved an accuracy of 100 % compared to the last one with 99.50 %. To reduce the transfer time, they have suggested using a TensorFlow lite framework destined for mobile to do a recognition without a needd cloud, contrary to the previous work."

- "Tiagrajah V (2019) *et al* published in the Proceedings of the 9th International Conference on System Engineering and Technology (ICSET) IEEEhave used Convolutional Neural Networks for object detection. They have presented a deep learning approach for robust detection of traffic light by comparing two object detection models and by evaluating the flexibility of the TensorFlow Object Detection Framework to solve the real-time problems. The models are Single Shot Multibox Detector (SSD) MobileNet V2 and Faster-RCNN. They showed that Faster-RCNN delivers 97.015%, which outperforms SSD by 38.806% for a model which had been trained using 441 images. They concluded that SSD shows more false negatives in object detection, but when used in mobile computing devices like Raspberry Pi, Android Devices, the inference time is faster in SSD compared to the Faster RCNN model."
- "Sumeet Sanjay Walam (2018) *et al* published in the Proceedings of the 2nd International Conference on Inventive Communication and Computational Technologies (ICICCT

ISSN 2454-9940
www.ijasem.org
Vol 18, Issue 4, 2024

2018) IEEE, proposes to detect and separate objects from a set based on their color. Proposed method of categorization is done on the basis of color of the object using a raspberry Pi 3. System categorizes the object into three different colors. The detection of the particular color is done by a light intensity camera to frequency convertor method."

3. *Proposed System*

Based on the articles, we decided to utilize TensorFlow lite as the framework and the SSD-mobilenet V2 model with a Raspberry Pi 3B+ for object detection. Based on the articles, we decided to utilize TensorFlow lite as the framework and the SSD-mobilenet V2 model with a Raspberry Pi 3B+ for object detection. The following considerations led to the decision:

- Object detection without deep learning algorithms, would mean thousands of lines of code which would be a cumbersome process. With lot of research going under Object Detection and lots of models becoming open-source, there will always be newer models in the future having better accuracy, lesser time for detection. But that is technological advancements, which needs to be dealt in a positive manner.

- A micro-controller like device with onsite computing capabilities will reduce the cost and time of the process by reducing the server latency time and also the need for employing servers for the same.

- An SSD-mobilenet based model gives better accuracy and lesser inference time compared to the other models. Based on the trade-off between time and accuracy, it is safer to select this model. In our project we try to detect flask, jug and cup and based on the detection result amessage is sent to the concerned room-service staff.

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT
ISSN 2454-9940
www.ijasem.org
Vol 18, Issue 4, 2024

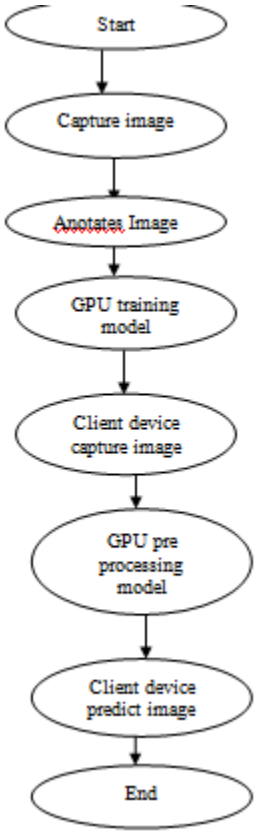


Figure 1 Flow chart

The billion of edge devices worldwide that are connected to the internet for a variety of purposes including data collection and analysis, are referred to as the Internet of Things (IoT). This edge devices can lower connectivity costs and preserve user privacy through data analytics. Computer vision, in particular, is a fundamental technology for developing such applications on these devices, where object-detection, the ability to identify different object types and mark their location is a crucial task. This paper describes a senior design project that used machine learning models to enable object recognition on a Raspberry Pi. It uses webcams and Raspberry Pi kits to run mobile deep learning models and identify predefined objects. Additionally, transfer learning techniques were used to refine the models on additional annotated datasets gathered from the internet in order to distinguish weapons in photos. In particular, it used a Google USN accelerator to speed up detection. The model can successfully identify both specified weapons on the Raspberry Pi and common items like people, laptops and keyboards according to preliminary validation data.

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT
ISSN 2454-9940
www.ijasem.org
Vol 18, Issue 4, 2024

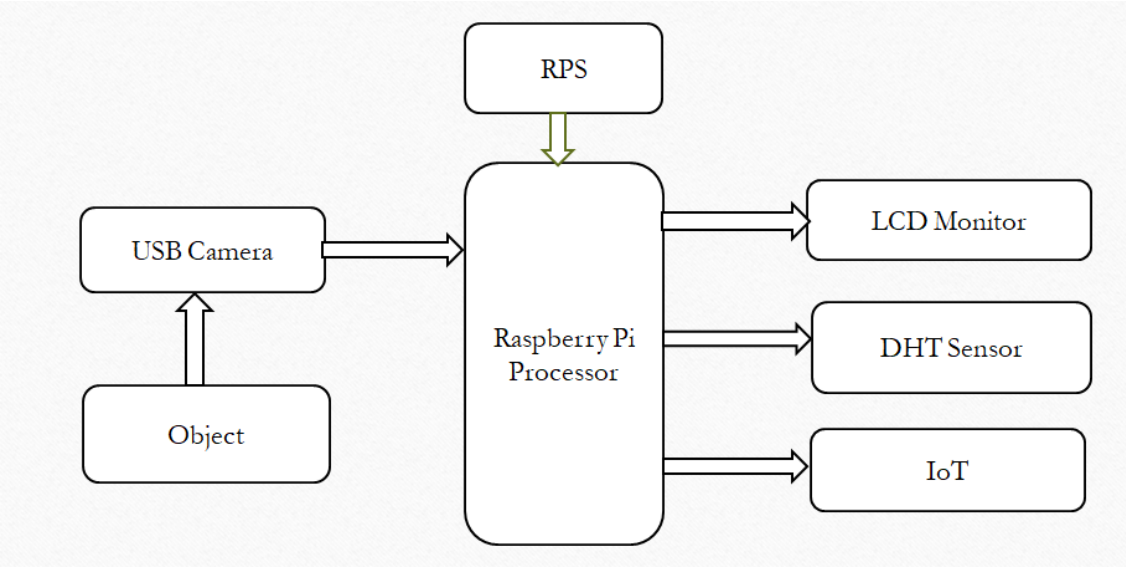


Figure 2 Block Diagram

- **TensorFlow:** All developers can use TensorFlow, an open source machine learning framework. Applications for deep learning and machine learning make use of it. TensorFlow was developed by the Google team in order to develop and investigate intriguing concepts related to artificial intelligence. All of Google's products use machine learning to enhance search engine performance, translation, picture captioning, and recommendation systems.It is designed to be executed on single or multiple CPUs and GPUs, making it a good option for complex deep learning tasks.

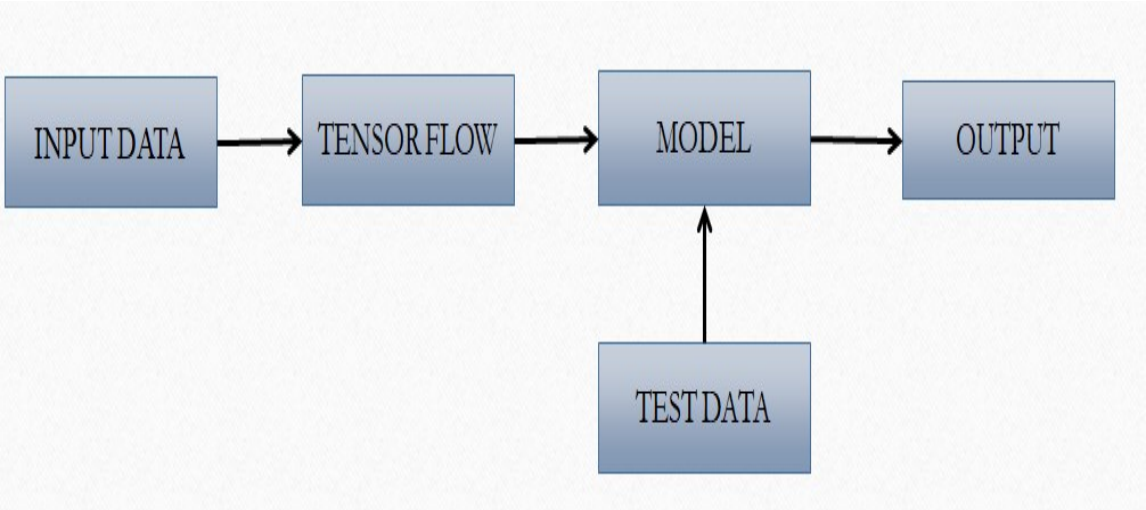


Figure 3 TensorFlow process

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT
ISSN 2454-9940
www.ijasem.org
Vol 18, Issue 4, 2024

Architecture of TensorFlow: There are three components to the TensorFlow architecture.

1. Preparing the data,
2. Developing the model,
3. Training it, and predicting it.

Gather images

For TensorFlow to train a decent detection classifier, an item must have at least hundreds of photos. To train a robust classifier, the training images should have a variety of backdrops and lighting condition, as well as the desired and random objects. The intended object should occasionally be partially hidden, overlapped with another object, or only halfway in the image. The three items we have for our project are the cup, flask, and jug. About 40 photos of each object were taken with our Android phone, and then another 60 photos containing several things were taken. To identify when things overlap, we are aware of this. We thus captured a lot of overlapping photos.

Figure 4 Gather images

In this as all our images size was more than 3 megabyte per image it took around 800 seconds to complete 1 step in the training process. So we decided to change all pictures size to 200 kilo bytes .To resize the images we used a script which is resizer.py .After we have all the pictures we need, move 20% of them to the \object_detection\images\test directory, and 80% of them to the \object_detection\images\train directory. Made sure there are a variety of pictures in both

ISSN 2454-9940
www.ijasem.org
Vol 18, Issue 4, 2024

the \test and \train directories.

Label pictures

With all the images gathered, it's time to label the desired objects in every picture. LabelImg [1] is a great tool for labeling images. This is a time-consuming process. LabelImg saves a .xml file containing the label data for each image. These .xml files will be used to generate TFRecords, which are one of the inputs to the TF trainer.

Once we have labeled and saved each image, there will be one .xml file for each image in the \test and \train directories.

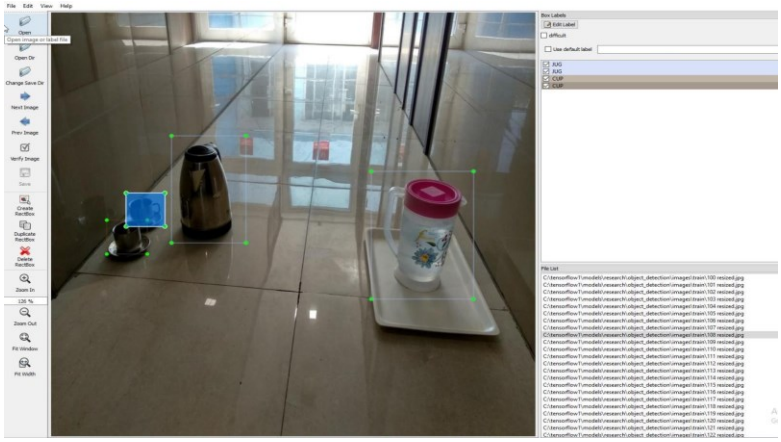


Figure 5 Label pictures

TFRecord: The performance of our import pipeline and the model,s training timecan be greatly impacted by storing our data in binary file format because we operate with enormous datasets. Binary data is more easier to read off the disk, requires less time to transfer, and takes up less disk space. In addition to its performance benefits, it connects with the library,s data import and preprocessing features and facilitates the blending of multiple datasets. This is advantageous, particularly for datasets that are too big to fit entirely in memory, as sequence data can be cached and will only be loaded from the disk when needed.

TensorFlow Lite Object Detection

An ideal framework for implementing lightweight deep learning models on edge devices with limited resources is TensorFlow Lite. TensorFlow Lite models can be utilized to achieve faster performance in real-time applications because they need less computing resources and have a faster inference time. After training a bespoke TensorFlow Object

ISSN 2454-9940
www.ijasem.org
Vol 18, Issue 4, 2024

Detection model, we now optimize its format for TensorFlow Lite to use and execute it on a Raspberry Pi.

Creating Optimized Tensorflow Lite Model

Since our detection model has previously been exported as a frozen graph for TensorFlow Lite, it must first pass through the TensorFlow Lite Optimizing Converter (TOCO) in order to be compatible with the TensorFlow Lite interpreter. In order for models to function well on TensorFlow Lite, TOCO translates them into an optimal Flat Buffer format. Before executing the model, we must additionally make a new label map. Therefore, this TOCO will use the following command to convert the resultant frozen graph (tflite_graph.pb) to the TensorFlow Lite flat buffer format (detect.tflite). Use this command to execute the floating model.

Tensorflow Obect Detection API on the Raspberry Pi

The following steps are done in Raspberry Pi to perform object detection API on live video feeds from a Pi camera or USB webcam. This module includes the object detection Pi camera.py script, which is a Python script that loads an object detection model in TensorFlow and uses it to detect objects in a Pi camera video feed. We use TensorFlow v1.8.0 on a Raspberry Pi Model 3B+ running on RaspbianStretch.

Step 1: Update the Raspberry Pi:

Step 2. Install TensorFlow.

Step 3. Install OpenCV.

Step 4. Compile and Install Protobuf.

Step 5. Set up TensorFlow Directory Structure and PythonPath Variable.

➢ Transfer Learning

Transfer learning (TL) is a research problem in machine learning (ML) that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem. For example, knowledge gained while learning to recognize cars could apply when trying to recognize trucks. This area of research bears some relation to the long history of psychological literature on transfer of learning, although formal ties

ISSN 2454-9940
www.ijasem.org
Vol 18, Issue 4, 2024

between the two fields are limited. From the practical standpoint, reusing or transferring information from previously learned tasks for the learning of new tasks has the potentialto significantly improve the sample efficiency of a reinforcement learning agent.

- **Edge Impulse:** Edge Impulse is the leading edge AI platform for collecting data, training models, and deploying them to your edge computing devices. It provides an end-to-end framework that easily plugs into your edge MLOps workflow.

Previously, we looked at edge_MLOps and how it can be used to standardize your edge AI lifecycle. This time, we introduce Edge Impulse as a platform for building edge AI solutions and edge MLOps pipelines.

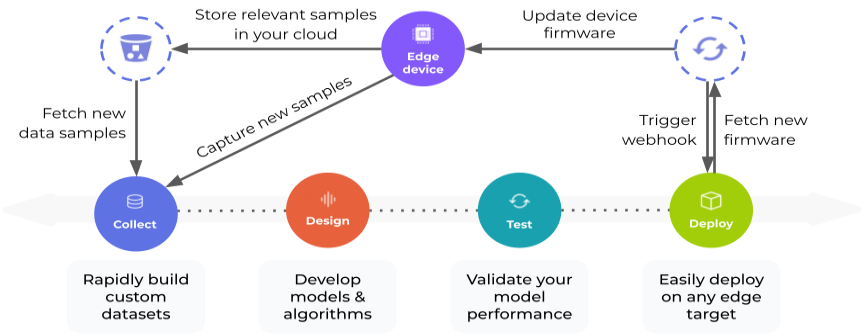


Figure 6 **Edge Impulse**

Steps involve in Edge Impulse Classifier

Step 1: Image Data Collection and Preprocessing.

Step 2: Training the image classifier model.

Step3: Real time object detection and classification..

Step4: Retraining.

Step5: Deploying the classifier to the edge device**.**

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT
ISSN 2454-9940
www.ijasem.org
Vol 18, Issue 4, 2024

Edge Impulse Studio

Edge Impulse Studio is a web-based tool with a graphical interface to help you collect data, build an impulse, and deploy it to an end device. Data can be stored, sorted, and labeled using the data acquisition tool.

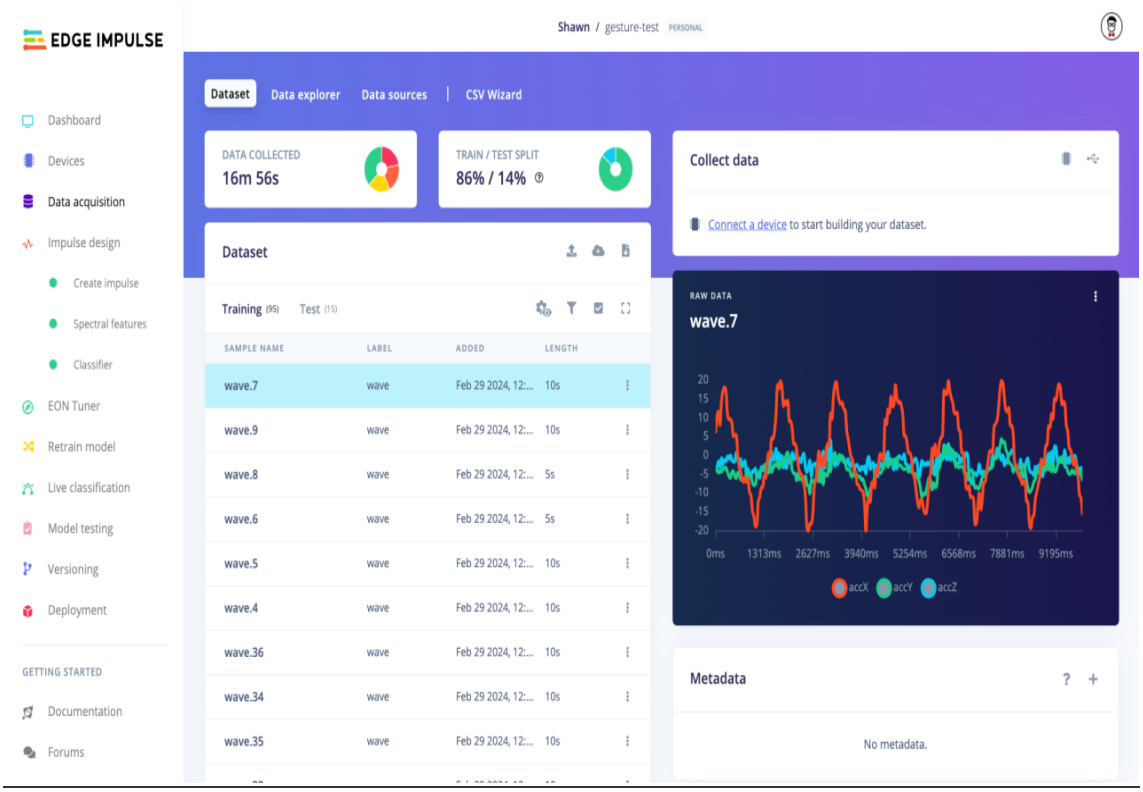


Figure 7 Edge Impulse data acquisition

From there, an impulse can be created that includes one or more feature extraction methods along with a machine learning model.

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT
ISSN 2454-9940
www.ijasem.org
Vol 18, Issue 4, 2024

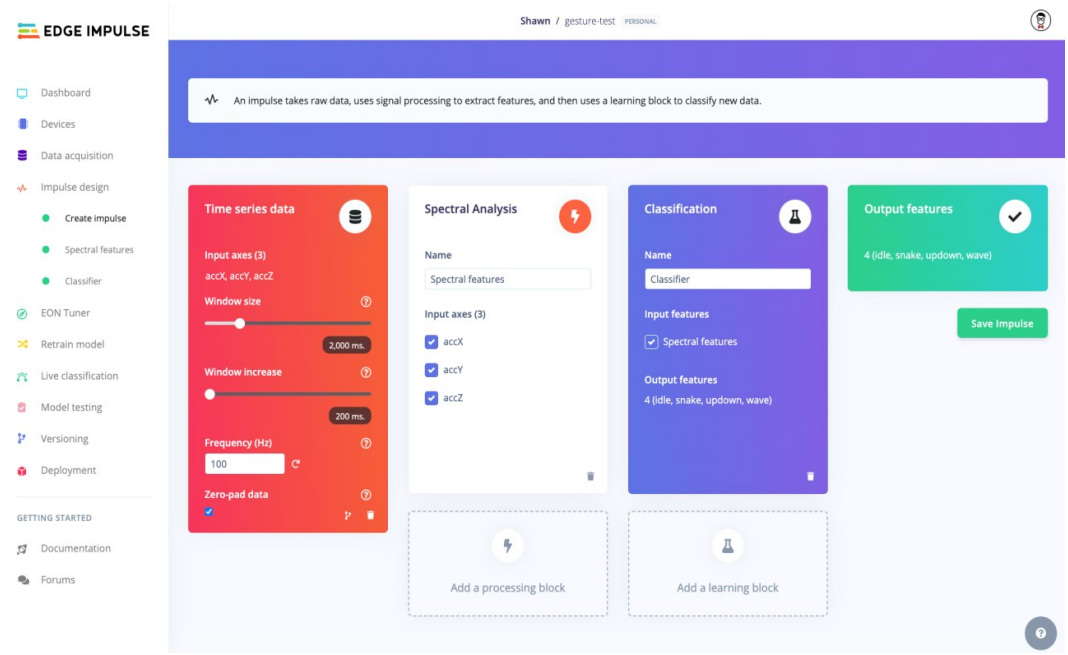


Figure 8 Edge Impulse design

A number of off-the-shelf feature extraction methods can be used and modified to suit the needs of your particular project. You can also design your own feature extraction method using a custom processing block.

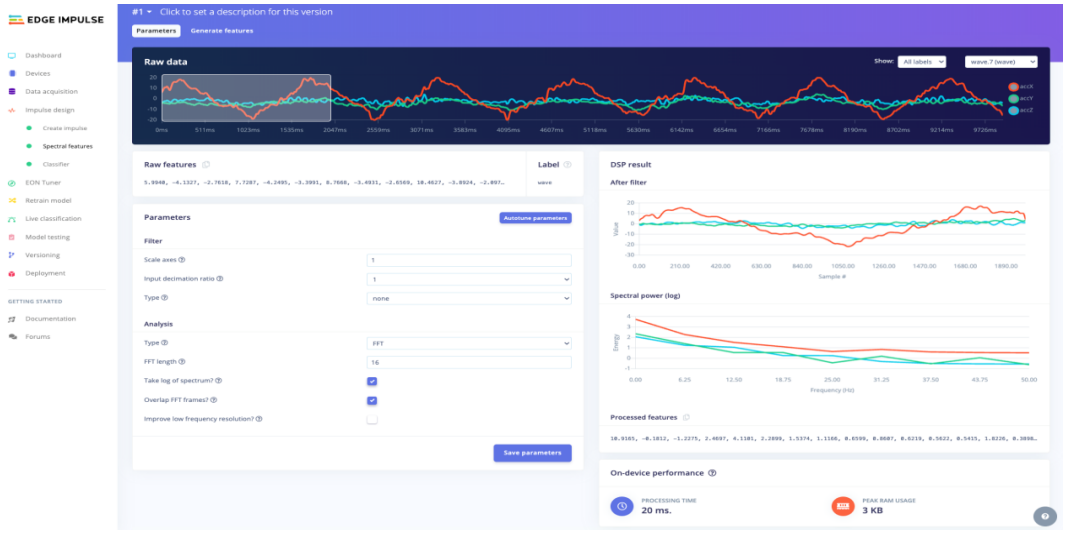


Figure 9 Edge Impulse processing block configuration

Next, you can train a machine learning model (including classification, regression, or anomaly detection) using a learning block. A number of pre-made learning

blocks can be used, but you can also create your own custom learning block or use the expert mode to modify the ML training code.

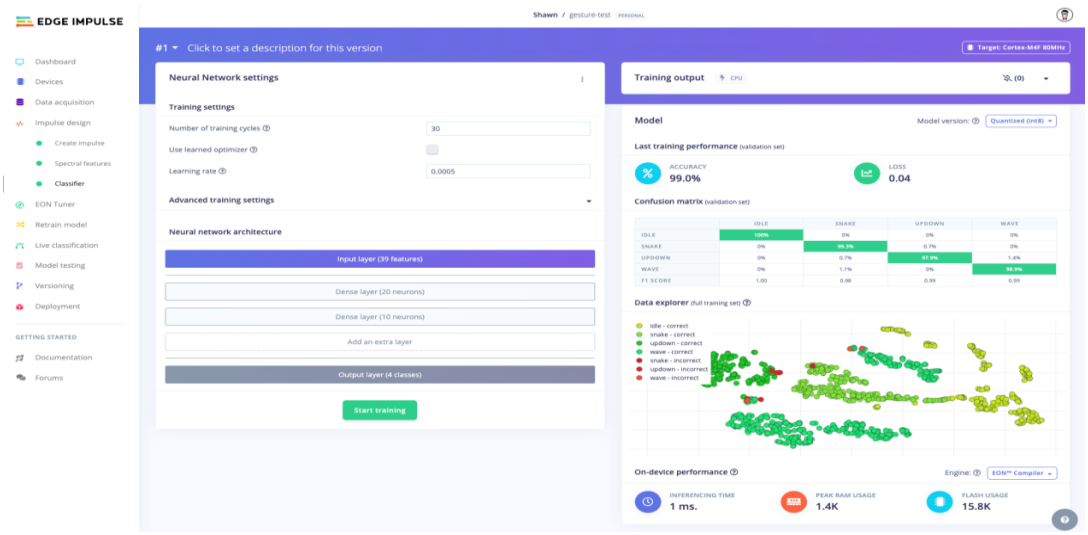


Figure 10 Edge Impulse learning block configuration

Once trained, the models can be tested using a holdout set or by connecting your device to ingest live data.

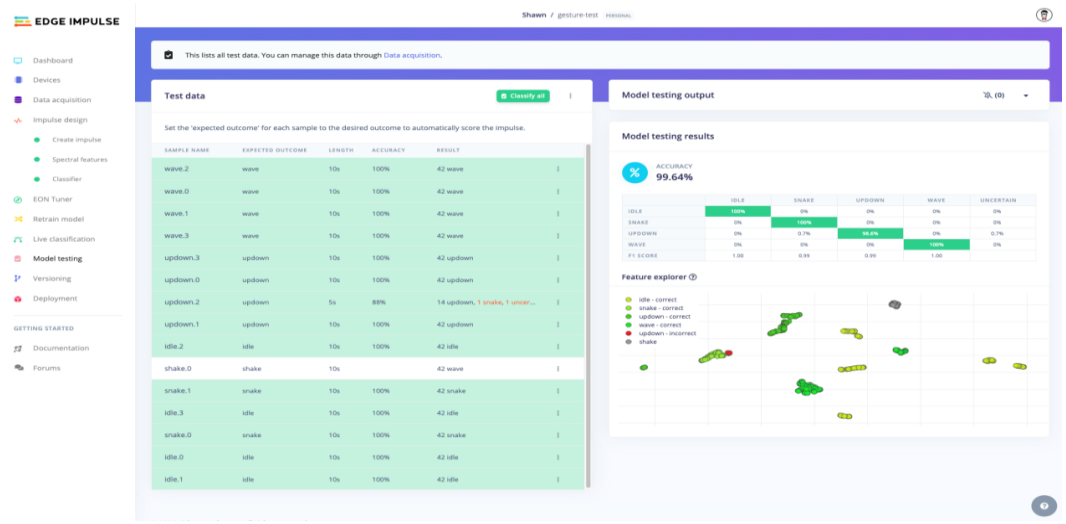


Figure 11 Edge Impulse testing

Finally, your full impulse can be deployed in a variety of formats, including a C++ library, Linux process (controlled via Python, Node.js, Go, C++, and others),

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT
ISSN 2454-9940
www.ijasem.org
Vol 18, Issue 4, 2024

Docker container, WebAssembly executable, or a pre-built firmware for supported hardware.

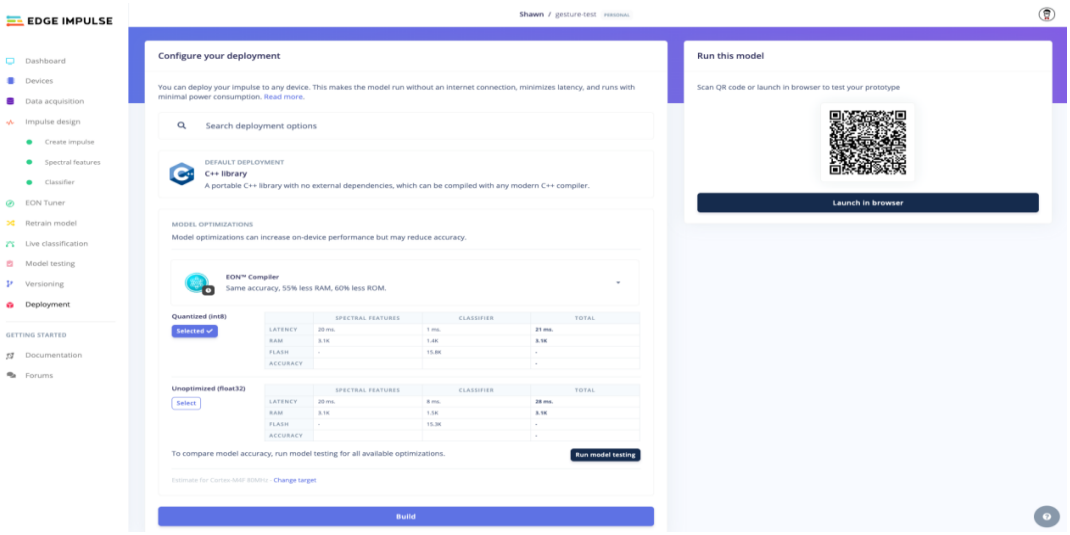


Figure 12 Edge Impulse deployment options

- **SSD_MobileNet_v2_coco_ model from TensorFlow's model zoo**

TensorFlow provides several object detections models (pre trained classifiers with specific neural network architectures) in its model zoo [1]. This model zoo provides a collection of detection models pre-trained on the COCO dataset, the Kitti dataset and the other things. Faster-RCNN models provide slower detection but higher accuracy, On the other hand, SSD-Mobile Net versions have a design that allows for quicker detection but lower accuracy.

We initially started with Faster-RCNN model and we are not able to convert the model in to tflite [1] for raspberry pi. So, we retrained our detector on the SSD-Mobile Net model and the detection worked considerably better.

Later on, we discovered that the SSD-Mobile Net model is the best way to use Object Detector on a device with little processing power, such a smart phone or Raspberry Pi.

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT
ISSN 2454-9940
www.ijasem.org
Vol 18, Issue 4, 2024

- **Neural Network (NN):** A neural network is a network or circuit of neurons, or in a modern sense, an artificial neural network, composed of artificial neurons or nodes. Thus a neural network is either a biological neural network, made up of real biological neurons, or an artificial neural network, for solving AI problems.

The connections of the biological neuron are modeled as weights. A positive weight reflects an excitatory connection, while negative values mean inhibitory connections. All inputs are modified by a weight and summed.

This activity is referred as a linear combination. Finally, an activation function controls the amplitude of the output. For example, an acceptable range of output is usuallybetween 0 and 1, or it could be −1 and 1.

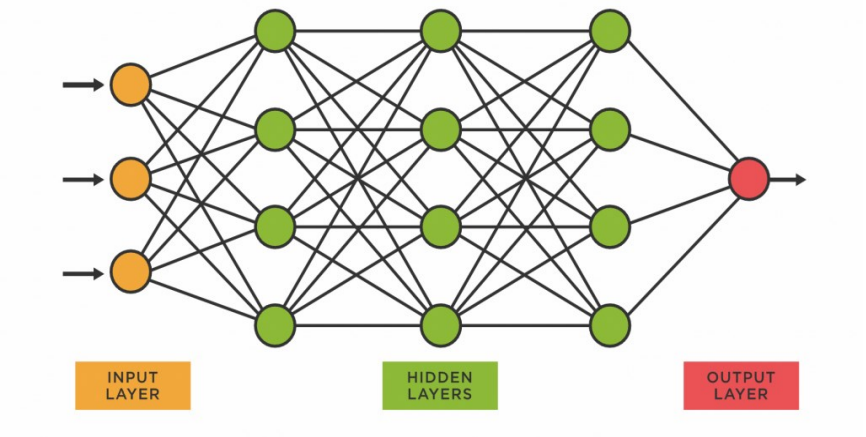


Figure 13 Architecture of Neural Network

- **Convolutional Neural Network (CNN)**: A convolutional network receives a normal color image as a rectangular box whose width and height are measured by the number of pixels along those dimensions, and whose depth is three layers deep, one for each letter in RGB. Those depth layers are referred to as channels.

As images move through a convolutional network, we will describe them in terms of input and output volumes, expressing them mathematically as matrices of multiple dimensions in this form: 30x30x30. From layer to layer, their dimensions change for reasons that will be explained below. Now, for each pixel of an image, the intensity of R, G and B will be expressed by a number, and that number will be an element in one of the

ISSN 2454-9940
www.ijasem.org
Vol 18, Issue 4, 2024

three, stacked two-dimensional matrices, which together form the image volume. Those numbers are the initial, raw, sensory features being fed into the convolutional network, and the ConvNets purpose is to find which of those numbers are significant signals that actually help it classify images more accurately. (Just like other feed forward networks wehave discussed).

Rather than focus on one pixel at a time, a convolutional net takes in square patches of pixels and passes them through a filter. That filter is also a square matrix smaller than the image itself, and equal in size to the patch. It is also called a kernel, which will ring a bell for those familiar with support-vector machines, and the job of the filter is to find patternsin the pixels.

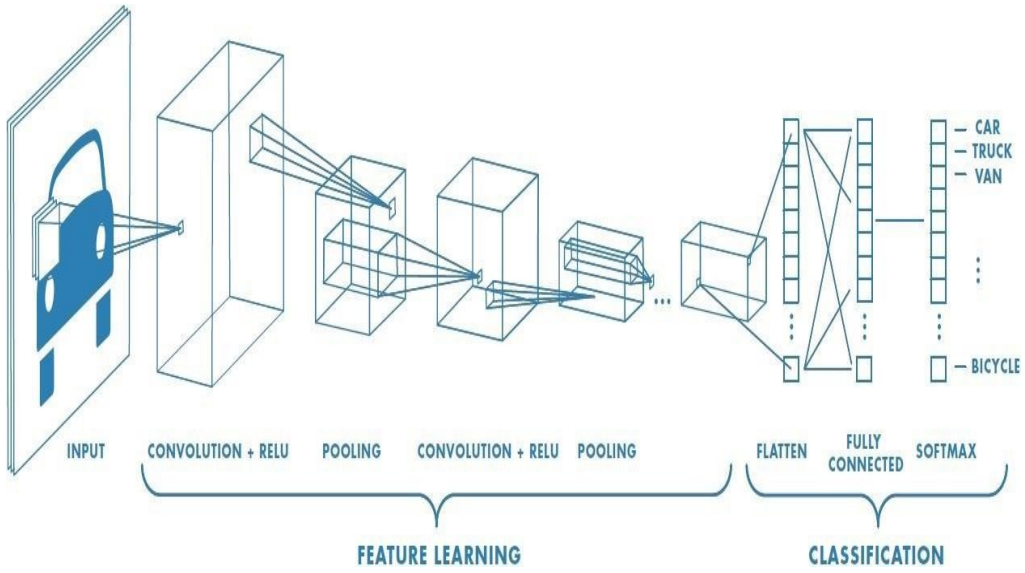


Figure 14 Architecture of Convolutional Neural Network

Computations in CNN: Before efficient CNN models, the computational cost of building blocks used in efficientCNN models, and performing convolution in spatial and channel domain is explained.

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT
ISSN 2454-9940
www.ijasem.org
Vol 18, Issue 4, 2024

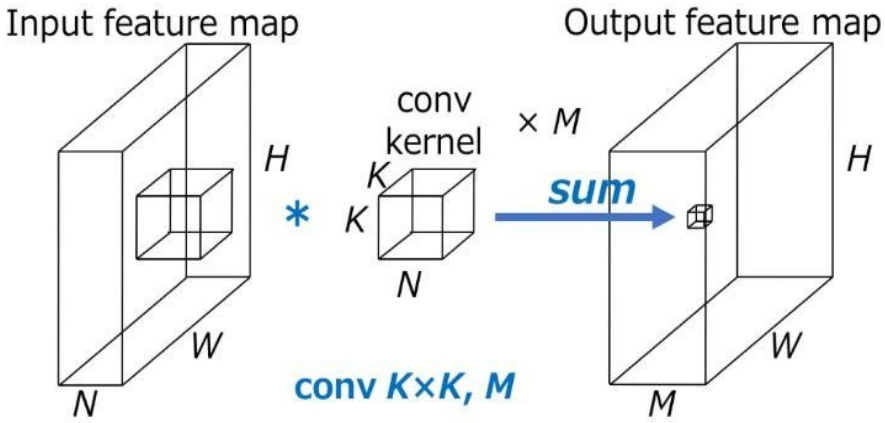


Figure 15 Computations in CNN

- HxW denotes the spatial size of output feature map,
- N denotes the number of input channels
- KxK denotes the size of convolutional kernel
- M denotes the number of output channels
- The computational cost of standard convolution becomes $HWNK^2M$ and it is proportional to
- The spatial size of the output feature map HxW
- The size of convolutional kernel K^2
- The number of input and output channels NxM

The above computational cost is required when convolution is performed on both spatial and channel domain. CNNs can be speeded up by factorizing this convolution.

Key Components of CNN:

1. Convolutional layers: Kernal, Stride, Padding.
2. pooling layers and
3. fully connected layers.

Kernel: Filter that was applied to the picture in order to extract the feature.

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT
ISSN 2454-9940
www.ijasem.org
Vol 18, Issue 4, 2024

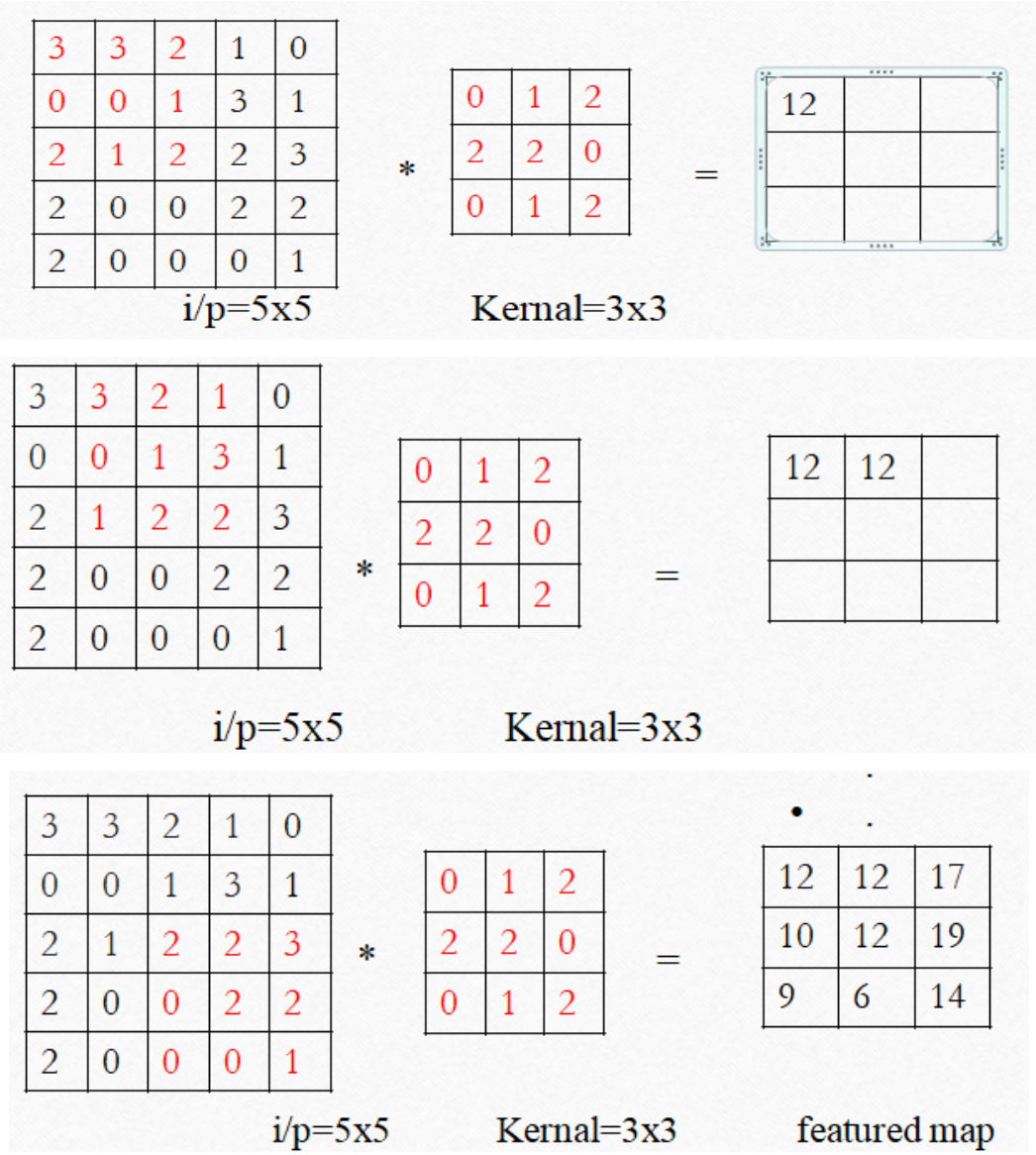


Figure 16 Kernal process

Stride: the degree of movement between the i/p image and the filter application.The filter is applied from top to bottom, left to right, and with a single pixel column change in both vertical and horizontal directions.

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT
ISSN 2454-9940
www.ijasem.org
Vol 18, Issue 4, 2024

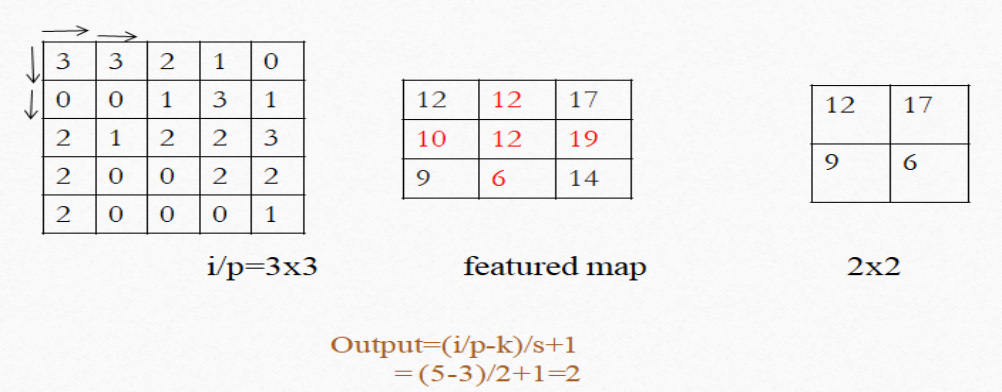


Figure 17 Stride process

- **Padding**: if i/p size is 5x5, then o/p should also be of size 5x5. so adding 0's in rows and columns.

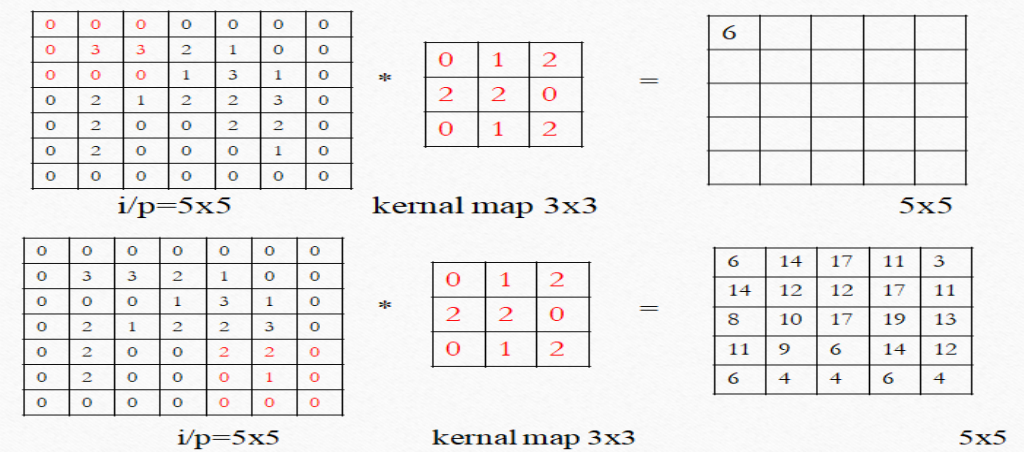


Figure 18 Padding process

- **Pooling**: In order to down sample the feature map's feature detection, pooling is necessary. Average pooling and maximum pooling are two popular pooling techniques that summarize the average and most activated feature presences.

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT
ISSN 2454-9940
www.ijasem.org
Vol 18, Issue 4, 2024

Max pooling:

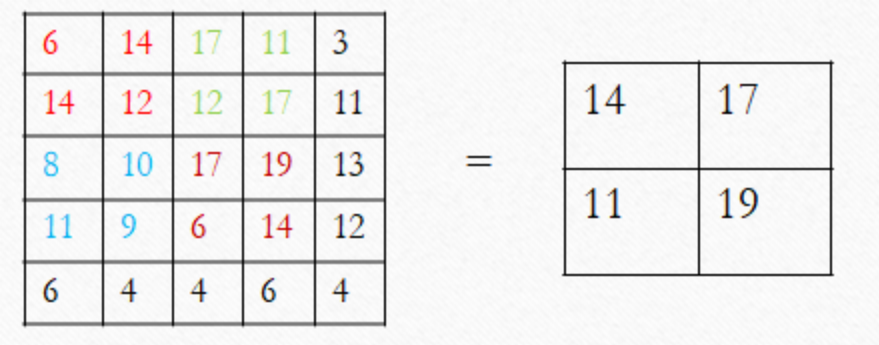


Figure 19 Max Pooling

Average pooling:

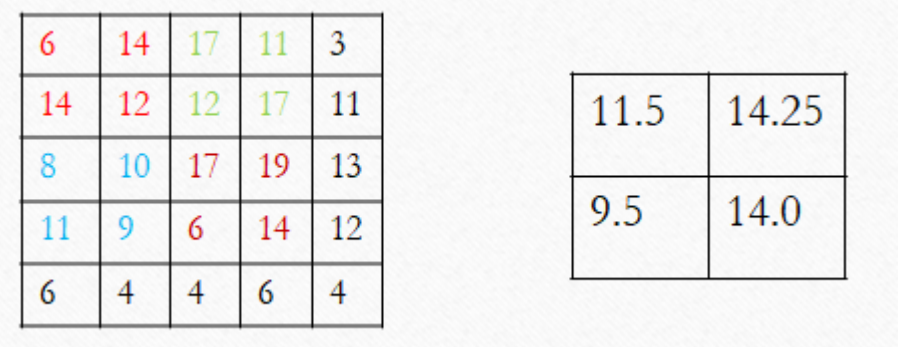


Figure 20 Average Pooling

- **Flattening**: creates a single column out of the full pooled featured map matrix, which is subsequently sent to NN for processing.

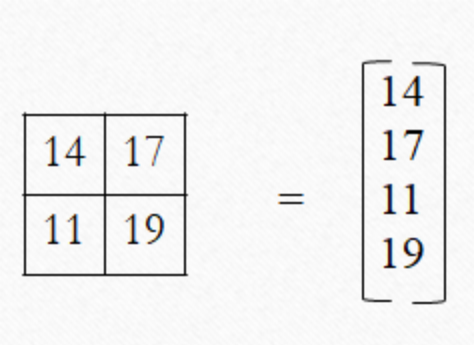


Figure 21 Flattening

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT
ISSN 2454-9940
www.ijasem.org
Vol 18, Issue 4, 2024

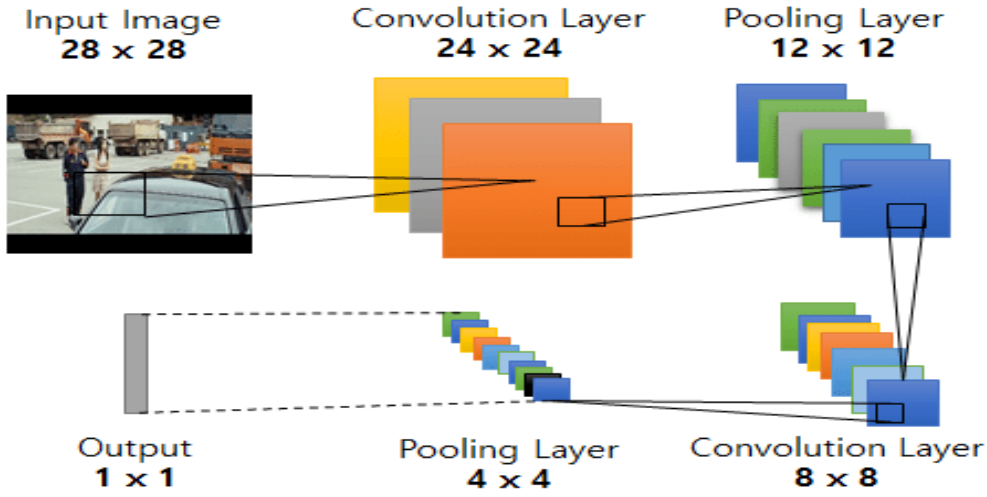


Figure 22 Complete process of CNN

➢ MOBILENET-V2

MobileNet-V2 is composed of an architecture conv1x1, conv3x3, and conv1x1. The first conv1x1 increases the dimensions of the input channel, then depth wise convolution performed. The final conv1x1decrese the dimension of the output channels.

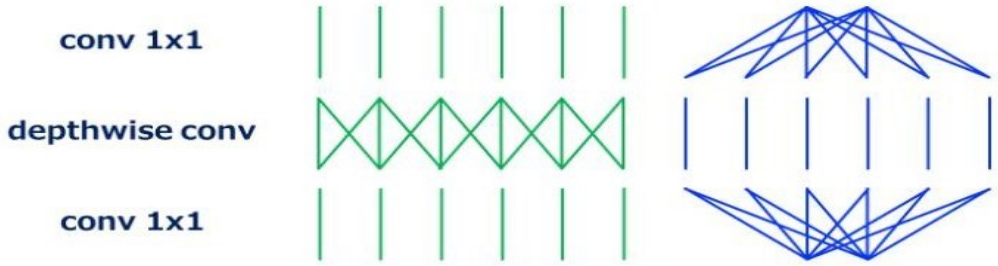


Figure 23 Convolution of MobileNet-V2

➢ DHT Sensor

The DHT (Digital Humidity & Temperature) sensor is a commonly used electronic component for measuring temeperature and humidity. It is popular due to its simplicity, low cost and ease of interfacing with microcontrollers like Arduino and Raspberry Pi. The DHT sensors comes in two types: DHT11 and DHT22, with DHT22 being more accurate and capable of measuring wide range of temperature and humidity.

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN 2454-9940

www.ijasem.org

Vol 18, Issue 4, 2024

4. *Results and Discussions*

- Once the script initializes (which can take up to 30 seconds), we will see a window showing a live view from our camera. Common objects inside the view will be identified and have a rectangle drawn around them.

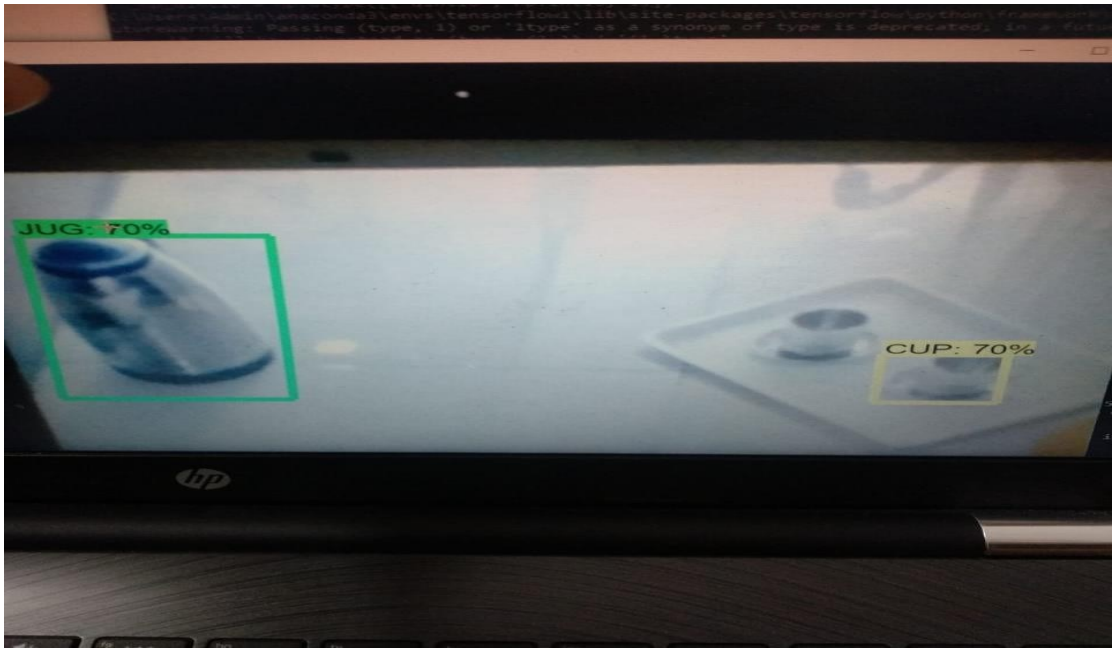
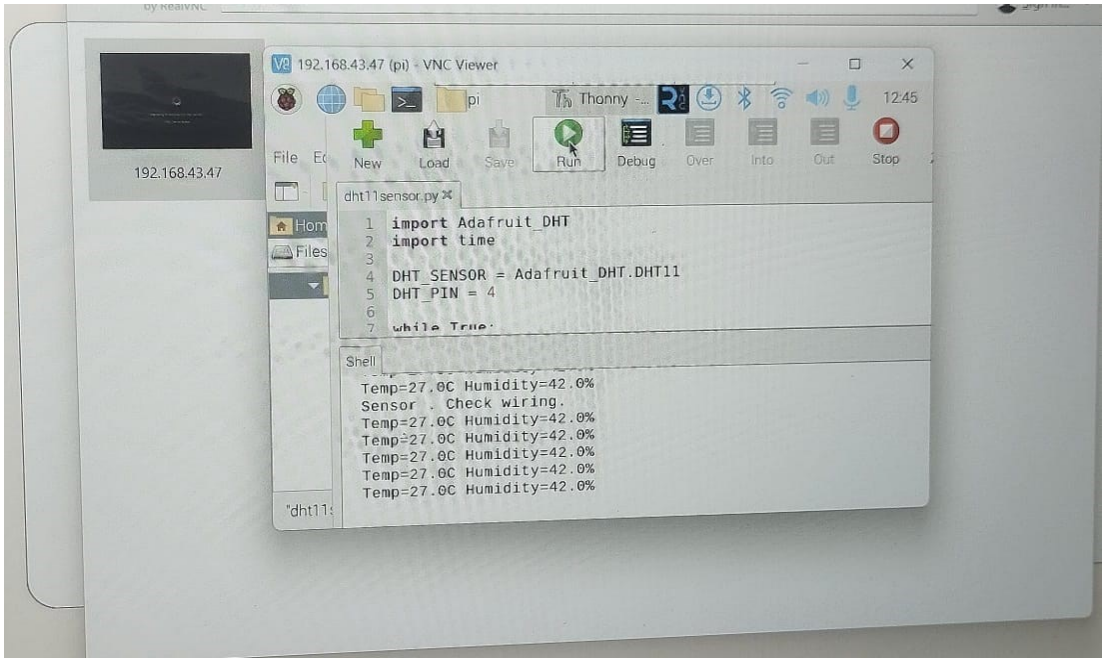


Figure 24 Final Output



ISSN 2454-9940
www.ijasem.org
Vol 18, Issue 4, 2024

Figure 25 DHT output

5. *Conclusion*

The proposed system was trained with 3 different kinds of datasets namely jug, cup and flask using SSDmobilenet model v2, and the test results were with 96% accuracy and 0.9 frames per second. Although the accuracy seems to be good, the frames per second is very low for a real time detection. In case of low traffic object detection where the objects are slow moving through the frame, then we can certainly consider using the Raspberry Pi for deep learning object detection. However, if we are developing an application that involves many objects that are fast moving, we should instead consider faster hardware.

Object detection is a key area of artificial intelligence that enables machines to perceive and interact with the physical world effectively.Combines object classification and localization to identify and locate objects in images or videos. Humidity & Temperature of particular area is known. CNN, and SSD with TensorFlow have significantly improved detection accuracy and efficiency. Using DHT sensor the dedicated areas Humidity and temoerature is calculated.

6. *References*

[1] George E.Sakr ,Maria Mokbel,Ali Hadi and Ahmad Darwich ,COMPARING DEEP LEARNING AND SUPPORT VECTOR MACHINES FOR AUTONOMOUS WASTE SORTING in 2016 IEEE International Multidisciplinary Conference on Electrical Technology (IMCET) ,vol 26,no.3,pp.657-853,March 2016

[2] Tiagarajah ,V.Janahiraman1 and Mohammad Shahrul Subuhan ,TRAFFIC LIGHT USING TENSORFLOW OBJECT DETECTION FRAMEWORK in 2017 IEEE 9th International Conference on System Engineering and Technology (ICSET),vol 27,no.4,pp.567-598,March 2017.

[3] Abdelhakim Zeroual ,P.F Rocha and M.P Silva,SSD MultiBox Object Detection ,published in the Proceedings of the International Conference on Networking and Advanced

ISSN 2454-9940
www.ijasem.org
Vol 18, Issue 4, 2024

Systems (ICNAS) IEEE,vol 25,no.2,pp.543-538,March 2018.

[4] Wei Liu,Dragomir Anguelov,Dumirtu Erhan,Christian Szegedy,"SSD:Single Shot MultiBox Detector " in arxiv Cornell University,2019.

[5] De Charette, R. and Nashashibi, F., "Traffic light recognition using image processing compared to learning processes," In 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, vol 15,no.6,pp.345-214,October 2009.

[6] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y. and Berg, A.C., "Ssd: Single shot multibox detector," in ArXiv,20 Dec 2014.

[7] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: "Single shot multibox detector",.In European Conference on Computer Vision.

[8] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg. "SSD: Single Shot MultiBox Detector". In ArXiv , 30 Mar 2018.

[9] https://www.raspberrypi.org/about/

[10] Python documentation,docs.python.org

[11] https://www.tensorflow.org/about/