Smart Surveillance using MobileNetV2

Dhanvanth P   
Electronics and Communication Engineering   
Sri Sairam Engineering college  
Chennai, India  
e8ec062@sairamtap.edu.in

Rajkumar S  
Electronics and Communication Engineering  
Sri Sairam Engineering CollegeChennai, India  
e8ec182@sairamtap.edu.in

Mrs. K. Subhashini  
Electronics and Communication Engineering  
 Sri Sairam Engieering College  
Chennai, Imdia  
subhashini.ece@sairam.edu.i*n*  
  
Aravind Siddharth L  
Electronics and Communication Engineering  
Sri Sairam Engineering CollegeChennai, India  
e8ec018@sairamtap.edu.in

*Abstract*—Surveillance systems have existed for long and have been executed with humans playing an integral part in the system. Hence, making the system obsolescent at the absence or heedlessness of the guard. According to studies, an average adult’s maximum attention span is 20 minutes. With the growing population and an alarming increase of the physical intrusion gathering the limelight, the human surveillance systems are in dire need of upgrade. Technology has evolved exponentially since the invention of the first surveillance system in 1969. Invention of IOT has made it possible to centralize a surveillance system, making the processing ability of the person assigned as the only limitation. With the recent advances in Convolutional Neural Networks and formulation of the CNN models, even a basic classification model capable of detecting a few threats can serve as a decent live surveillance system.

Keywords—Surveillance, object detection, MobileNetV2, IoT, Realtime-surveillance

# Introduction

In the area of surveillance, object detection is required to have a strong protection against any danger. Most of the surveillance systems make use of cc cameras to monitor the area and they need someone to check the camera’s feed of the monitored area constantly. It is a very difficult process for people who have to secure sparsely populated areas like restricted areas, roads which cannot be monitored consistently by a person. Object detection using deep learning can be used to secure such places even without the help of a person. [1], [2]

## C*onvolutional Neural Network (core technology)*

Convolutional Neural Networks (CNNs) are a form of feed-forward artificial neural network. These are used to analyze and manipulate visual imagery and graphs. A CNN architecture is a stack of distinct layers that transform the input volume to an output volume through a differentiable function. Biological neurons interact with one another and produce outputs based on inputs. CNN’s architecture was loosely influenced by such neurons, also known as perceptrons. They have recently gained popularity as a result of recent technological advances and computational capabilities that allow the processing of large quantities of data and the training of sophisticated algorithms in a reasonable amount of time.[5]

## MobileNetV1

MobileNets are a class of efficient models, used for mobile and embedded vision applications. It is based on a streamlined architecture which uses depth-wise separable convolutions to build light weight deep neural networks. Depth wise Separable Convolution is used to reduce the model size and complexity. One of the major uses of MobileNets are mobile and embedded vision applications. The key features of MobileNets are that they are small, low-latency, low-power models. They are parameterized to meet the resource constraints of a varied range of use cases.

1. Smaller model size: Fewer number of parameters
2. Smaller complexity: Fewer multiplication and additions

Two parameters are introduced so that MobileNet can be tuned easily: one is Width Multiplier and the other is Resolution Multiplier. These simple global hyper-parameters efficiently tradeoff between latency and accuracy. Based on the constraints of the problem, hyper-parameter tuning feature allows the model builder to choose the right sized model for their use case.[4]

MobileNet could be used in object detection, fine grain classification, face recognition, large-scale geo localization, etc,.

## MobileNetV2

MobileNetV2 is a great improvement over MobileNetV1 in terms of classification, semantic segmentation, and object identification for mobile visual recognition. It builds upon the ideas from MobileNetV1. It uses depth wise separable convolution as its efficient building blocks. The two new features that were introduced in the MobileNetV2 architecture includes:

1. Linear bottlenecks between the layers
2. Shortcut connections between the bottlenecks

The MobileNetV2 architecture delivers highly accurate results, while keeping the parameters and mathematical operations as low as possible. This helps us to bring deep neural networks to mobile devices. Overall, MobileNetV2 models are faster for the same accuracy across the entire latency spectrum. In particular, the new models use 2x fewer operations, need 30% fewer parameters and are about 30-40% faster on a Google Pixel phone than MobileNetV1 models, all while achieving higher accuracy.[6]

For object detection and segmentation applications, MobileNetV2 poses as a very effective feature extractor. For example, for detection when paired with the newly introduced SSD Lite the new model is about 35% faster with the same accuracy than MobileNetV1.

## MobileNetV2 Layer Architecture

1. MobileNetV2 layer architecture

|  |  |  |  |
| --- | --- | --- | --- |
| ***Type / Stride*** | | ***Filter Shape*** | ***Input Size*** |
| Conv / s2 | | 3 x 3 x 3 x 32 | 224 x 224 x 3 |
| Conv dw / s1 | | 3 x 3 x 32 dw | 112 x 112 x 32 |
| Conv / s1 | | 1 x 1 x 32 x 64 | 112 x 112 x 32 |
| Conv dw / s2 | | 3 x 3 x 64 dw | 112 x 112 x 64 |
| Conv / s1 | | 1 x 1 x 64 x 128 | 56 x 56 x 64 |
| Conv dw /s1 | | 3 x 3 x 128 dw | 56 x 56 x 128 |
| Conv / s1 | | 1 x 1 x 128 128 | 56 x 56 x 128 |
| Conv dw / s2 | | 3 x 3 x 128 dw | 56 x 56 x 128 |
| Conv / s1 | | 1 x 1 x 128 x 256 | 28 x 28 x 128 |
| Conv dw / s1 | | 3 x 3 x 256 dw | 28 x 28 x 256 |
| Conv / s1 | | 1 x 1 x 256 x 256 | 28 x 28 x 256 |
| Conv dw / s2 | | 3 x 3 x 256 dw | 28 x 28 x 256 |
| Conv / s1 | | 1 x 1 x 256 x 512 | 14 x 14 x 256 |
| 5 x | Conv dw / s1  Conv / s1 | 3 x 3 x 512 dw  1 x 1 x 512 x 512 | 14 x 14 x 512  14x 14 x 512 |
| Conv dw / s2 | | 3 x 3 x 512 dw | 14 x 14 x 512 |
| Conv / s1 | | 1 x 1 x 512 x 1024 | 7 x 7 x 512 |
| Conv dw / s2 | | 3 x 3 x 1024 dw | 7 x 7 x 1024 |
| Conv / s1 | | 1 x 1 x 1024 x 1024 | 7 x 7 x 1024 |
| Avg Pool / s1 | | Pool7 x 7 | 7 x 7 x 1024 |
| FC / s1 | | 1024 x 1000 | 1 x 1 x 1024 |
| Softmx / s1 | | Classifier | 1 x 1 x 1000 |

## MobileNetV2 Layer Metrics

1. MobileNetV2 layer metrics

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Input*** | ***Operator*** | ***t*** | ***c*** | ***n*** | ***s*** |
| 2242 x 3 | conv2d | - | 32 | 1 | 2 |
| 1122 x 3 | bottleneck | 1 | 16 | 1 | 1 |
| 1122 x 1 | bottleneck | 6 | 24 | 2 | 2 |
| 562 x 24 | bottleneck | 6 | 32 | 3 | 2 |
| 282 x 32 | bottleneck | 6 | 64 | 4 | 2 |
| 142 x 64 | bottleneck | 6 | 96 | 3 | 1 |
| 142 x 96 | bottleneck | 6 | 160 | 3 | 2 |
| 72 x 160 | bottleneck | 6 | 320 | 1 | 1 |
| 72 x 320 | Conv2d 1 x 1 | - | 1280 | 1 | 1 |
| 72 x 128 | avgpool 7 x 7 | - | - | 1 | - |
| 1 x 1 x 1280 | conv2d I x 1 | - | k | - | - |

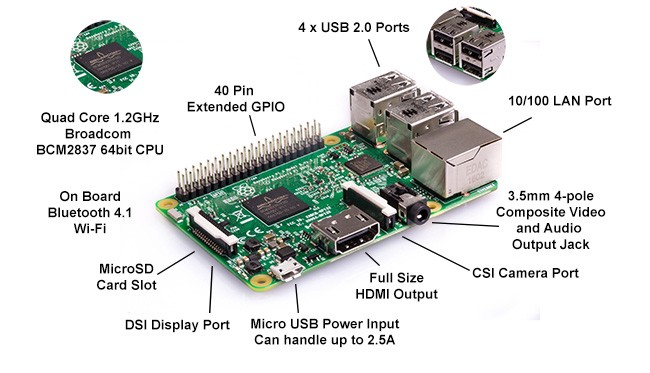
# Literature Review

Surveillance systems is being widely used quite ever and innovations within the field follow an equivalent. There has been quite few research and systems proposed focused on surveillance. A previous system proposed in **Human detection in surveillance videos using MobileNet** **(Bouafia Yassine, Guezouli Larbi, Lakhlef Hicham)** used a MobileNet deep convolution neural network with transfer learning approach to create deep learning model for human classification. We achieved an honest accuracy and comparative precision, where the learned features are extracted automatically. They provide most competitively accurate leads to image recognition tasks but they have more computing power and enormous space memory which is challenging for embedded devices.

As suggested in the paper **IOT based urban surveillance using Raspberry Pi and Deep learning with Mobile-Net Pre-trained model (G.SreeHarsha, Yogapriya J, Sathya Vignesh R, Vaishnavi.R.G, B.HariKrishnaReddy & G.Aravind)**, the object detection is required to possess a stronger protection within the surveillance areas. a number of the surveillance systems uses cc cameras to watch the world. It needs someone to see the output feed especially with-out rest. It is a difficult process for people that need to secure distant areas, like fields and restricted areas, which can't be monitored consistently by an individual. Object detection using raspberry pi and deep learning with pre-trained model can be readily used to secure the areas even without the need for a person to take care of the monitoring. [3], [7], [8]

# Hardware

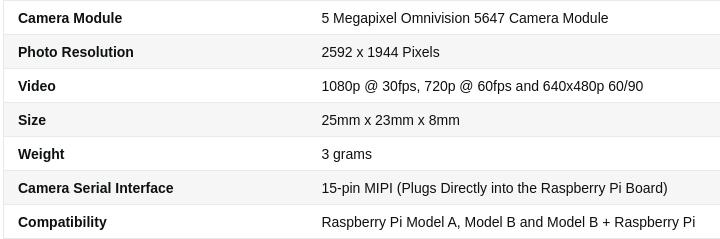
## Raspberry pi



1. Raspberry pi module

* Raspberry Pi (*figure 1*) is an efficient credit card-sized computer that is cost-effective and portable, and used in embedded system for application specific software implementations. Raspberry Pi foundation to enlighten and empower computer science teaching in schools and other developing countries. Since its creation, various open-source communities have been contributing to open-source apps, operating systems, and various other small form factor computers similar to Raspberry Pi. Raspberry Pi is a full-fledged Computer that can be considered for almost all computing-intensive tasks. [9]

## Camera module



1. Camera module specifications

A fixed lens with a resolution of 5 megapixels is integrated with the camera module’s board. This camera board can capture with a resolution of 2592 x 1944 pixels. It is capable of recording high quality videos that supports 1080p @ 30fps, 720p @ 60fps and 640x480p 60/90 format.

## Benchmark

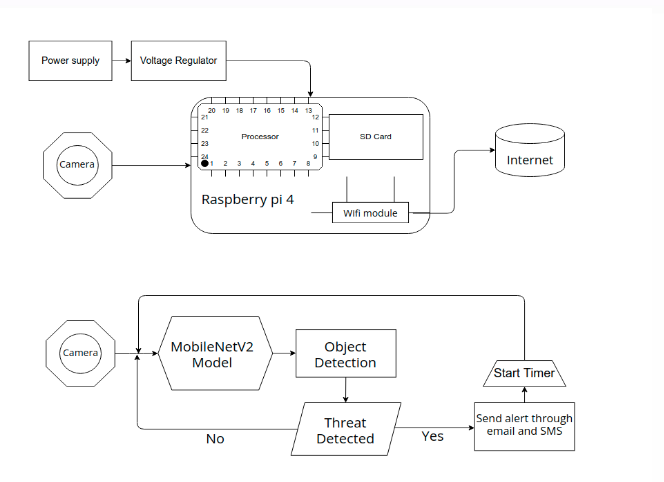
1. Benchmark of Various models

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Network*** | ***Top 1*** | ***Params*** | ***Mul-Adds*** | ***CPU*** |
| MobileNetV1 | 70.6 | 4.2M | 575M | 113ms |
| ShuffleNet (1.5) | 71.5 | 3.4M | 292M | - |
| ShuffleNet (x2) | 73.7 | 5.4M | 524M | - |
| NasNet-A | 74.0 | 5.3< | 565M | 183ms |
| MobileNetV2 | 72.0 | 3.4M | 300M | 75ms |
| MobileNetV2 (1.4) | 74.7 | 6.9M | 585M | 143ms |

* Executing a surveillance system can require a moderate processing power (lesser number of parameters as depicted in Table III) and it can be difficult to afford for small business or an average household, the Raspberry Pi 4 has enough processing to run a surveillance system, monitor for threats and is affordable compared to other surveillance systems.
* A Raspberry Pi is capable of getting up to ~0.9 frames per second when applying deep learning for object detection with Python and OpenCV which is fast enough for a surveillance system and using multiprocessing, the system can become much more efficient.
* Raspberry Pi temperatures approach, but not exceed, the 80°C point where thermal throttling of the CPU would occur during inferencing using TensorFlow and TensorFlow Lite models.
* Benchmarks using the AI2GO platform and the binary weight network models shows inferencing time competitive with the NVIDIA Jetson Nano using their TensorRT optimized models.

# Proposed model

Our proposed model is a smart surveillance system that runs with no necessary human intervention and constantly monitors the area. The system will alert the user when a threat is detected and classified depending on the classification and nature of threat, concerned authorities may be alerted. The surveillance system’s is workflow is represented in *figure 3*.



1. Block Diagram

## Approach

As shown in *Figure 3*, The system consists of a Raspberry pi with a camera module the raspberry pi that can be connected to external monitor or a device capable of using RDP (Remote Desktop Protocol) or VNC (Virtual Network Computing) can be used to connect securely to the device. The raspberry pi will be constantly monitoring the area with camera module while using OpenCV to parse the input from the camera module on which detection is done and the threat is classified. After classification of threat, if a threat is detected then the user is alerted with threat classification, type and nature through a SMS while the image of the same will be send through email.

# Experiment

## Setup

The model has 2 types of layers. The first layer is called a **depth**-**wise convolution** (performs lightweight filtering by applying a single convolutional filter per input channel) and the second layer is a 1 x 1 convolution, known as **pointwise convolution**, that is used to build new features. Overall, the model has a combination of 3 conv layers and 7 bottleneck layers. Typically, it has a computational cost of 300 million multiply-adds and uses 3.4 million parameters.

A dataset of 20k images belonging to all 10 classes were used to train the model in batches of 96 images. The video frames are pre-processed using OpenCV library and then sent to the model as input. The model outputs the class and the box boundaries. These output values are matched with the training dataset’s labels and then using differential equation, the weights are adjusted accordingly. Thus, the model was trained.

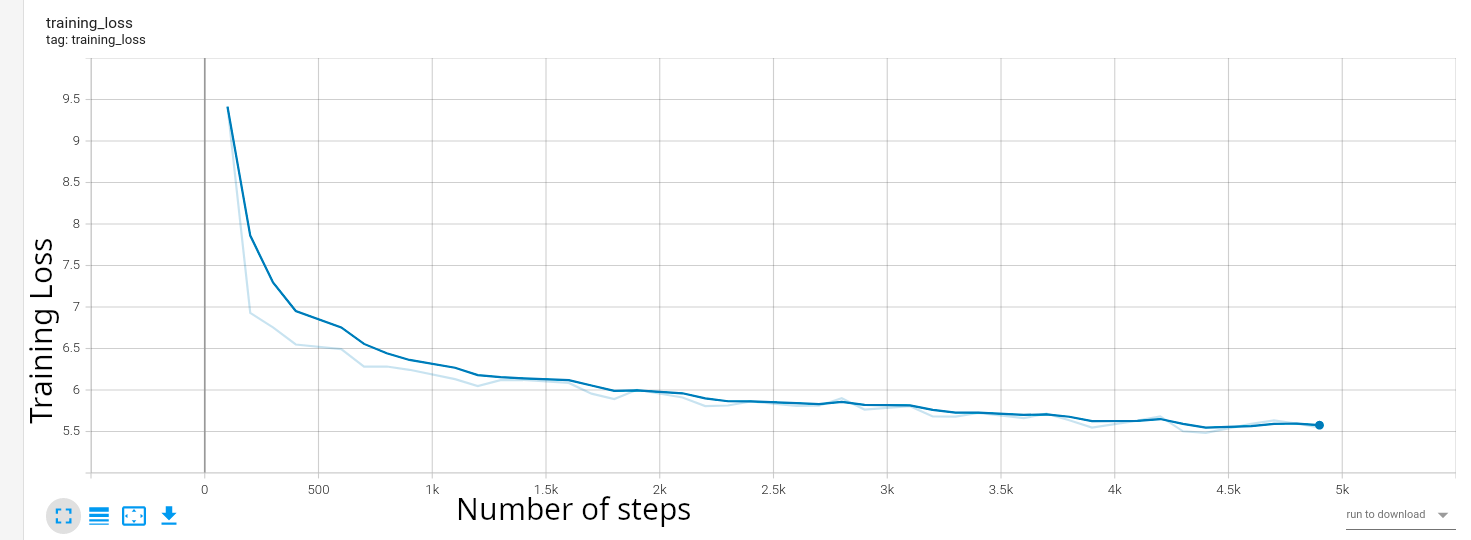
## Testing

The model was loaded onto the SD card of the raspberry pi module and the main script was run. After setting up the credentials for receiving alerts, the program starts to retrieve input feed from the camera. The model was given a real time footage by placing the camera in front of a house, to monitor the movement on the road. The model was thus evaluated for its accuracy over all the frames.

# Observation

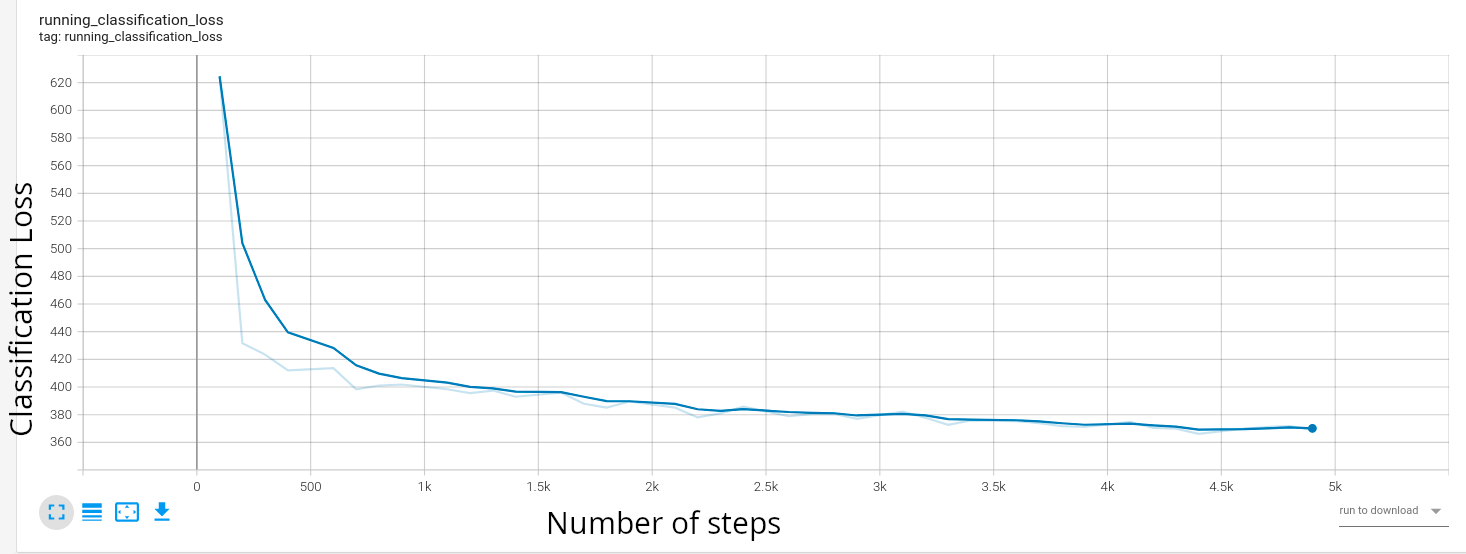
## Training

The classification loss is plotted to train the classify head for determining the type of target class (represented in *figure 4*).



1. Classification loss

The training loss is a metric used to assess the error of the model on the training set (represented in *figure 5*).



1. Training loss

## Testing

The trained MobileNetV2 model was tested against both the open images dataset as well as sample videos with weapons in them (as a real weapon could not be showcased in front of the camera) and the comparison is depicted in *Table IV*.

1. Tabulation

|  |  |  |
| --- | --- | --- |
| ***Metrics*** | ***MobileNetV1*** | ***MobileNetV2*** |
| Precision | 0.75 | 0.79 |
| Recall | 0.72 | 0.697 |
| F1 score | 0.734 | 0.74 |
| Accuracy | 0.84 | 0.88 |

##### Conclusion

Both the training and implementation of the image dataset in the proposed threat detection model will be split into two classes: passive and threat. Based on the object detection and localization techniques used, the OpenCV deep neural network should produce positive results in terms of accuracy and speed. The above proposed method is special in that it uses the MobilenetV2 image classifier to not only classify images, but also identify and bound their locations with minimal computation, resulting in more accurate and efficient performance in low-end mobile devices. Many previous studies were able to improve accuracy with their dataset. However, since the dataset was gathered from a variety of other sources and the images used in the dataset were manually cleaned to improve the accuracy of the results, the case of obtaining of multiple wrong predictions will have been successfully eliminated. The proposed implementation using MobilenetV2 model should, assist in dealing with the problem of constant and reliable surveillance system in the neighborhood.

##### References

1. Verizon Small Business Essentials Resources. 2022. From Edison to Internet: A History of Video Surveillance - Verizon Small Business Essentials Resources. [online] Available at: <https://www.verizon.com/business/small-business-essentials/resources/edison-internet-history-video-surveillance-105000047/#:~:text=1969%3A%20The%20first%20video%20home,in%20the%20non%20government%20market.> [Accessed 11 March 2022]..
2. En.wikipedia.org. 2022. Closed-circuit television - Wikipedia. [online] Available at: <https://en.wikipedia.org/wiki/Closed-circuit\_television#Technology>.
3. R, S., G, V., Aravind, G., SreeHarsha, G., HariKrishnaReddy, B. and J, Y., 2021. IOT based urban surveillance using RaspberryPi and Deep learning with Mobile-Net Pre-trained model. [online] Ejmcm.com. Available at: <https://ejmcm.com/article\_1997.html> [Accessed 3 December 2021].
4. A. G. Howard, G. Andrew, et al. “MobileNets: Efficient convolutional neural networks for mobile vision applications”, arXiv preprint arXiv: 1704.04861, 2017.
5. Karen Simoyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. CoRR, abs/1409.1556, 2014.
6. GitHub. 2022. GitHub - d-li14/mobilenetv2.pytorch: 72.8% MobileNetV2 1.0 model on ImageNet and a spectrum of pre-trained MobileNetV2 models. [online] Available at: <https://github.com/d-li14/mobilenetv2.pytorch>
7. Patil, Rahul & Srinivas, Ram & Yelubenchi, Rohith & Vinay, N.R. & PRATIBA, D.. (2017). IoT Enabled Video Surveillance System Using Raspberry Pi. 1-7. 10.1109/CSITSS.2017.8447877..
8. Ijeat.org. 2022. [online] Available at: <https://www.ijeat.org/wp-content/uploads/papers/v10i4/D24040410421.pdf> [Accessed 13 March 2022]..
9. Nayyar, Anand & Puri, Vikram. (2015). Raspberry Pi-A Small, Powerful, Cost Effective and Efficient Form Factor Computer: A Review. International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE). 5. 720-737.