PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

100-ft Ring Road, Bengaluru – 560 085, Karnataka, India

Report on

Automated Garbage Segregator

Submitted by

Ajaya Bharathwaj (01FB15EEC017)

Ananya Deoghare (01FB15EEC030)

Anusha Shenoy (01FB15EEC044)

January - April 2019

under the guidance of

Dr Koshy George

Professor

Department of ECE

PES University

Bengaluru -560085

FACULTY OF ENGINEERING

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

PROGRAM B.TECH.

CERTIFICATE

This is to certify that the Report entitled

Automated Garbage Segregator

is a bona fide work carried out by

Ajaya Bharatwaj (01FB15EEC017)

Ananya Deoghare (01FB15EEC030)

Anusha Shenoy (01FB15EEC044)

In partial fulfillment for the completion of 8th semester course work in the Program of Study B.Tech. in Electronics and Communication Engineering, under rules and regulations of PES University, Bengaluru during the period Jan – Apr. 2019. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report. The report has been approved as it satisfies the 8th semester academic requirements in respect of project work.

Signature with date & Seal	*Signature with date & Seal*	*Signature with date & Seal*
(Dr Koshy George)	*Dr. Anuradha M*	*Dr. B. K. Keshavan*
Internal Guide	*Chairperson*	*Dean - Faculty of Engg. &Technology*

Name and signature of the examiners:

1.

2.

3.

DECLARATION

We, Ajaya Bharathwaj, Ananya Deoghare and Anusha Shenoy**,** hereby declare that the report entitled, ***Automated Garbage Segregator,*** is an original work done by us under the guidance of **Dr Koshy George**, *Professor*, ECE Department and is being submitted in partial fulfillment of the requirements for completion of 8th Semester course work in the Program of Study B.Tech. in Electronics and Communication Engineering.

PLACE: Bangalore
DATE: 6th May 2019

NAME AND SIGNATURE OF THE CANDIDATES

1. Ajaya Bharathwaj

2. Ananya Deoghare

3. Anusha Shenoy

Acknowledgment

The satisfaction and euphoria that accompany the accomplishment of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement provided motivation and crowned our efforts with success.

Our most sincere and grateful acknowledgment to PES University for providing us with the opportunity to pursue our degree and thus helping us to shape our careers. We are highly indebted to our project guide, Dr. Koshy George, Professor of the Department of Electronics and Communication Engineering for his continuous valuable guidance, advice and persistent encouragement throughout the duration of the project.

We would also like to extend our gratitude to Prof Anuradha M, Head of the Department of Electronics and Communication for her constant encouragement and support. We would be remiss if we did not express our gratitude to all the teaching and nonteaching staff and management of PES University, Bengaluru for their assistance and cooperation.

Abstract

In this paper, we proposed to build a crane which can automatically segregate waste into biodegradable, non-biodegradable and electronic waste. For simplicity, we have considered only 2 examples for each garbage class. The main aim is to extract images of the garbage and pass it through a deep network. Based on the output obtained from the deep network, the crane picks up the object and puts it in the respective bin. We also compared different Machine Learning approaches for classifying to find out which network works best for our dataset.

Table of contents

List of Tables and Figures

List of Tables

List of Figures

1. Introduction

1.1 Motivation

In our country, managing waste is one of the biggest issues. Our government has come up with a national campaign called 'Swachh Bharat' to increase awareness of this issue. Waste is easier to dispose of if it is segregated.

The biodegradable wastes can be converted into manure which can then be used in farms and domestic gardens to improve the quality of the soil. Electronic waste (consisting of various electronic parts) can be reused and recycled. The non-biodegradable waste can be either recycled or disposed in a more efficient manner depending on the quality and type of waste. If the waste is not segregated it is usually dumped in landfills or on the roads and it makes its way into the water-bodies or stomachs of other living organisms. It attracts rodents and other disease-causing bacteria, chokes drain, the land becomes barren. Also, the economic value of waste is best realized when it is segregated. Currently, there is no such system of segregation of dry, wet and metallic wastes at the household level. So we need to come up with efficient techniques to segregate waste.

1.2 Organisation of the report

The rest of the report is structured as follows. In Chapter 2, we list out some of the recently published works that explore various waste segregation techniques. Chapter 3 explains the datasets used. Chapter 4 presents the software aspect of the project, which includes various feature extraction methods and Convolutional Neural Networks. Chapter 5 gives the details of the hardware built. Chapter 6 explains the integration between the software and hardware part of the project. Chapter 7 is the final chapter in which the results and conclusions are present.

2. Literature Survey

This chapter will briefly explain the existing literature available and provide an overview of other related works.

Many people have worked with the idea of garbage segregation. Fitzwatler G. Ang et al[2], have made a line following robot which is able to pick up the desired bin and collect the waste in the bin. Their problem statement involved sorting steel cans, aluminium cans, glass bottles and plastic bottles using various sensors placed on the conveyor belt for sorting. Magnet motors were used to remove steel cans, metal detectors were used for aluminium cans, capacitive sensors for glass bottles and plastic bottles would travel down the conveyor belt which would end in a basin. This model achieved an accuracy of 80 - 90 %.

Aleena V.J et al[3] also worked on segregating garbage but on a more diverse dataset. The garbage, in this case, was segregated as Metallic, organic and dry. The metallic consisted of Safety pins, paper clips, batteries, nails. The organic waste consisted of kitchen waste, leftover food, vegetable peel, rotten fruits/vegetables. The dry waste consisted of paper, small bottles, heavy cartons, milk cover, dry leaves, clothes, tetra packs. They also placed their garbage on a conveyor belt. The metallic waste was collected using electromagnets, a blower was used to remove all the plastic/dry waste and the rest of the waste which was left at the conveyor belt was classified as organic waste. They achieved an accuracy of 80%.

Rashmi M. Kittali et al [6] used PLC (Programmable Logic Controller) for the segregation of garbage into wet and dry waste. They used various sensors on the conveyor belt for the segregation of garbage. They were positioned on the position of the garbage with respect to the hydraulic sensors below them, and this was fed as an input to the PLC. They used IR sensors to detect the presence of an object on the conveyor belt. Moisture sensors were used to segregate wet waste from dry waste. Metal detection sensors / Inductive proximity sensors were used to sense the metallic waste. Plastic detection sensor was used to detect the different sized plastic bottles /objects.

Kavya M et al [10] use the method of pushing the waste onto the conveyer belt, the presence of waste is first identified by the use of Infra-red sensor at the start end of the conveyor belt, the waste moves further for detection with inductive sensor to detect it is metal. If it is detected metal, electromagnet rotates to in a direction to collect the metallic waste. Then demagnetization takes place and waste is dropped into a dustbin. Further conveyer is moved and the dry waste is blown out using air blower. Here light particles like plastic, paper, etc. get segregated. Conveyer belt moves further, now the moisture sensor detects the wet waste and it is dropped into another dustbin.

As we notice, most of the solutions revolve around the same mechanism. The other authors have placed or dropped the garbage on a conveyor belt. Different sensors were used by them to detect if the object is present on the conveyor belt. They used magnets/metal sensors to detect metallic waste, moisture sensors to detect wet waste, blower / capacitive sensors to detect dry waste.

In our project, we have not used conveyor belts or magnets or any sensors for garbage detection. We use the image of the garbage to classify the data. The object is picked from the heap/waste is kept at a particular position, the claw is then used to pick up the object, the camera then clicks a picture of the object. The deep network is then used to classify the object which was picked up.

3. Dataset

3.1 Caltech-256 Dataset

The Caltech-256 object category dataset is a challenging set of 256 object categories containing a total of 30,607 images. It is an extension of the Caltech-101 dataset with more number of categories. It also consists of a 257th category called Clutter category for testing background rejection. For our project, we have excluded the 257th category.

The minimum number of images per category in Caltech-256 is 80. The average number of images range from 80 to 300 per category. Certain categories like clutter have 800+ images. (Also, the face has around 500 images, planes have around 800). The images are not of the same size. The size varies from around 4kB to 50kB. Top view, front view, all side views are also taken into account. The images have different backgrounds. For certain objects (example. bag-packs), things are of different shapes and sizes so as to cover maximum types of that object. Some categories have 2d representations. Things which have a lot of different styles and textures have a number of images, while those types which have the same texture and shape have less number of images.

Caltech 256 gives a very varying range of images for each category, which proves to be one of the most challenging datasets. This ensures that the training data is large and diverse, ensuring better training for any machine learning algorithm.

The images are not always focused on the object, due to which accuracies obtained using this dataset is not very high. It consists of very random classes, for example, Superman, Jesus Christ, Baseball Bat, Kayak, Simpsons, cockroach, etc.

Initially to test our network we have considered only 4 classes from the Caltech-256 dataset. The classes considered were PCI-Card, Saturn, Soda-can, and Superman. These classes were chosen at random.

3.2 Our garbage Dataset

Our garbage dataset consists of 3 categories namely Biodegradable, Non-Biodegradable, Electronic Waste. For simplicity, we have considered only 2 classes under each category.

- Biodegradable - Banana peel and paper cups
- Non-Biodegradable - Cans (Crushed) and plastic Boxes
- Electronic - Circuit board and Computer Mouse

We have created the dataset by taking some pictures from the internet, and some pictures from various cameras. The images are present in various orientations. The dataset consists of RGB images of size 224x224. The images under each category have various backgrounds and multiple objects in various orientations.

TABLE 3.2: Garbage Dataset

Serial Number	Waste Category	Class
1	Biodegradable	Banana Peel Paper Cup
2	Non-Biodegradable	Crushed Cans Plastic Boxes
3	Electronic Waste	Circuit Board Computer Mouse

The dataset is divided into train, validation and test/check for training the deep network.

- Train :

  The images in train are used to train the weights of the neural network and update the weights depending on the performance of the network.

  We have used 70 images per class for training. Hence, the total number of images under train is 420.

- Validation :

  The images in validation are used to determine the performance of the network during it's training based on the validation accuracy it gives.

  We have used 5 images per class for validation. Hence, the total number of images under validation is 30.

- Test :

  The images in test are used to determine the performance of the network after it is trained. It is used to find the accuracy of the network.

  We have considered 10 images per class. Hence, the total number of images under test is 60.

Hence the total number of images in the dataset is 510.

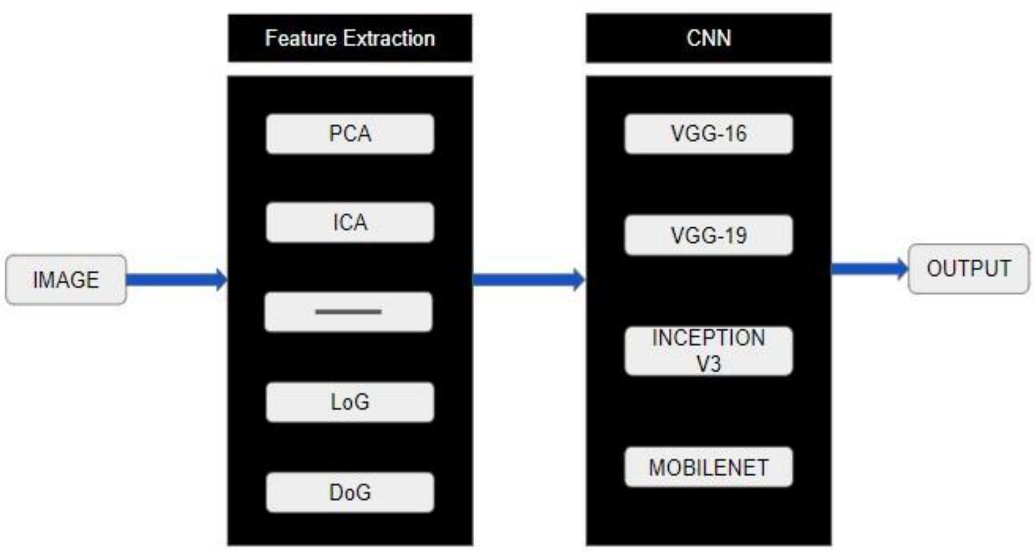
4. Software

4.1 Introduction

A combination of various Feature Extraction techniques and Convolutional Neural Networks were tested on 4 classes of Caltech-256. The Feature Extraction techniques used were Principal Component Analysis (PCA), Independent Component Analysis (ICA), Laplacian of Gaussian (LoG), Difference of Gaussian (DoG). The output images from these feature extraction techniques were passed through different CNNs and the results were compared. We also compared these results with no feature extraction done on images.

We tested these various combinations on just 4 classes of Caltech-256 dataset, namely PCICard, Saturn, Soda-Can, and Superman. The network and feature extraction method which showed good accuracy, was also tested on the entire Caltech-256 dataset, just to see how well the network performs. The best network was then trained on our garbage dataset for real-time classification of the garbage.

Fig. 4.1 Classification Techniques used



4.2 Feature Extraction

Feature Extraction is the process of extracting essential features/pixels of an image for improved and faster classification. It also reduces the computational complexity and the space required to store information/input/image as the number of features is drastically reduced when compared with the original image since the redundant features are removed.

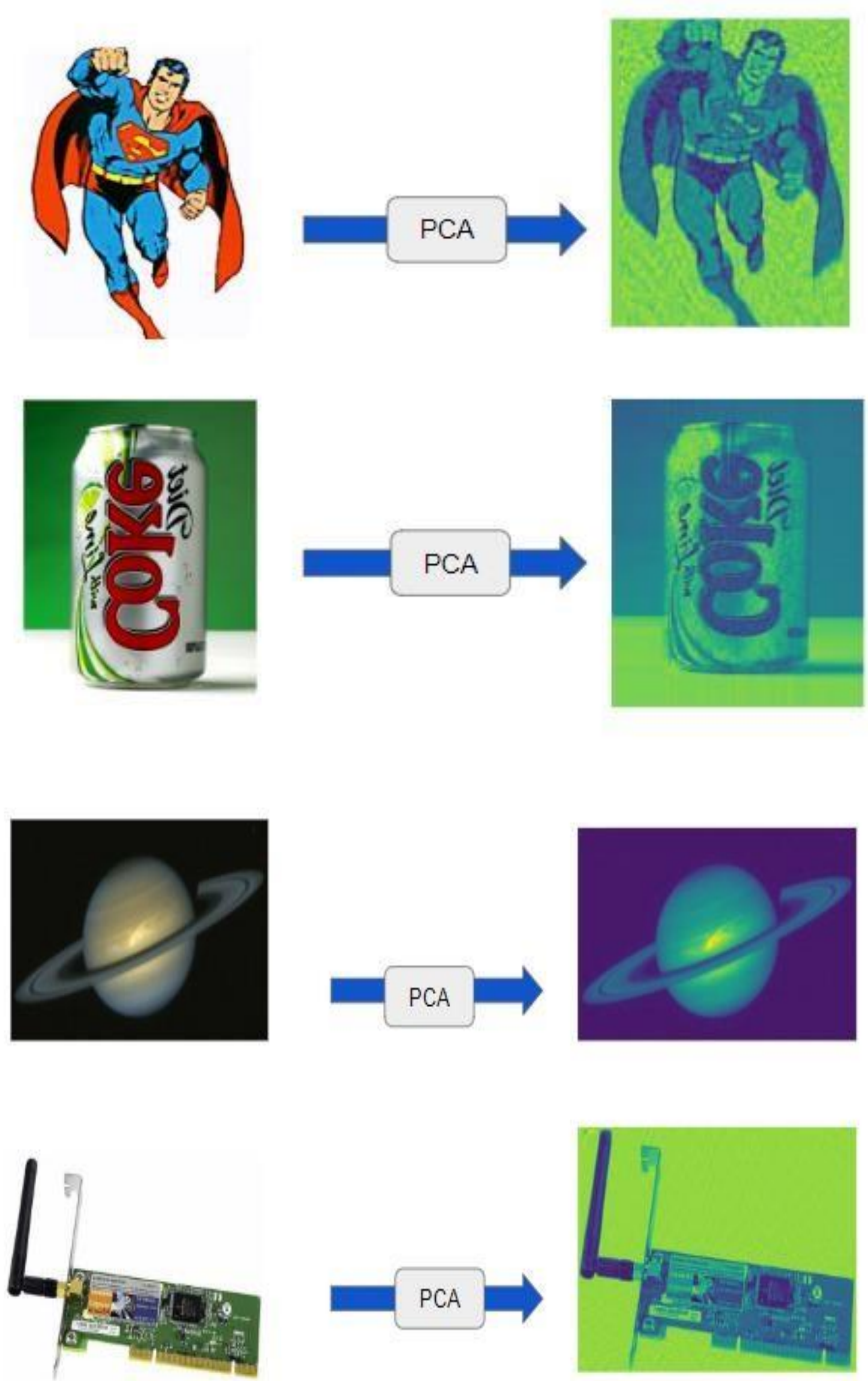
The following 4 feature extraction techniques were used on the input images:

- Principal Component Analysis (PCA)
- Independent Component Analysis (ICA)
- Laplacian of Gaussian (LoG)
- Difference of Gaussian (DoG)

4.2.1 Principal Component Analysis (PCA)

- PCA is a statistical technique. It uses an orthogonal transformation that converts a set of variables which are correlated into a set of linearly uncorrelated variables. These uncorrelated variables are called principal components.
- Principal Components are orthogonal to each other and are ordered in such a manner that the variation retained in the original variables becomes less as we move down the order. Thus, the variation retained in the 1st Principal component is the maximum variation that was present in the original set of variables and the 2nd principal component retains the 2nd maximum variation, and so on.
- T = X W is the transformation that maps a data vector x(i) from the initial vector space of s variables to a new space of s variables. Also, we don't need all principal components, hence we keep only the 1st n principal components corresponding to the 1st n eigenvectors (where these n eigenvectors are the eigenvectors of the covariance matrix of X) leading to the transformation TL = X WL , where the matrix TL has only n columns.
- PCA learns a linear transformation $\mathbf{t} = \mathbf{W}^T \mathbf{x}$, where $\mathbf{x} \in \mathbf{R}^p$, $\mathbf{t} \in \mathbf{R}^L$, where the columns of matrix W form an orthogonal basis for the L features that are decorrelated.
- The principal components are eigenvectors of the covariance matrix of the original input (dataset). They correspond to the direction with the greatest variance in data in the original space.
- Each eigenvector has an associated eigenvalue. The eigenvalue is a scalar that indicates how much variance there is in the data along the direction of the principal component.
- We applied PCA with 25 and 50 components on the images from the 4 classes that we considered from Caltech-256.
- On applying PCA, a vector of reduced dimension is obtained. To obtain the image back, inverse PCA is applied.

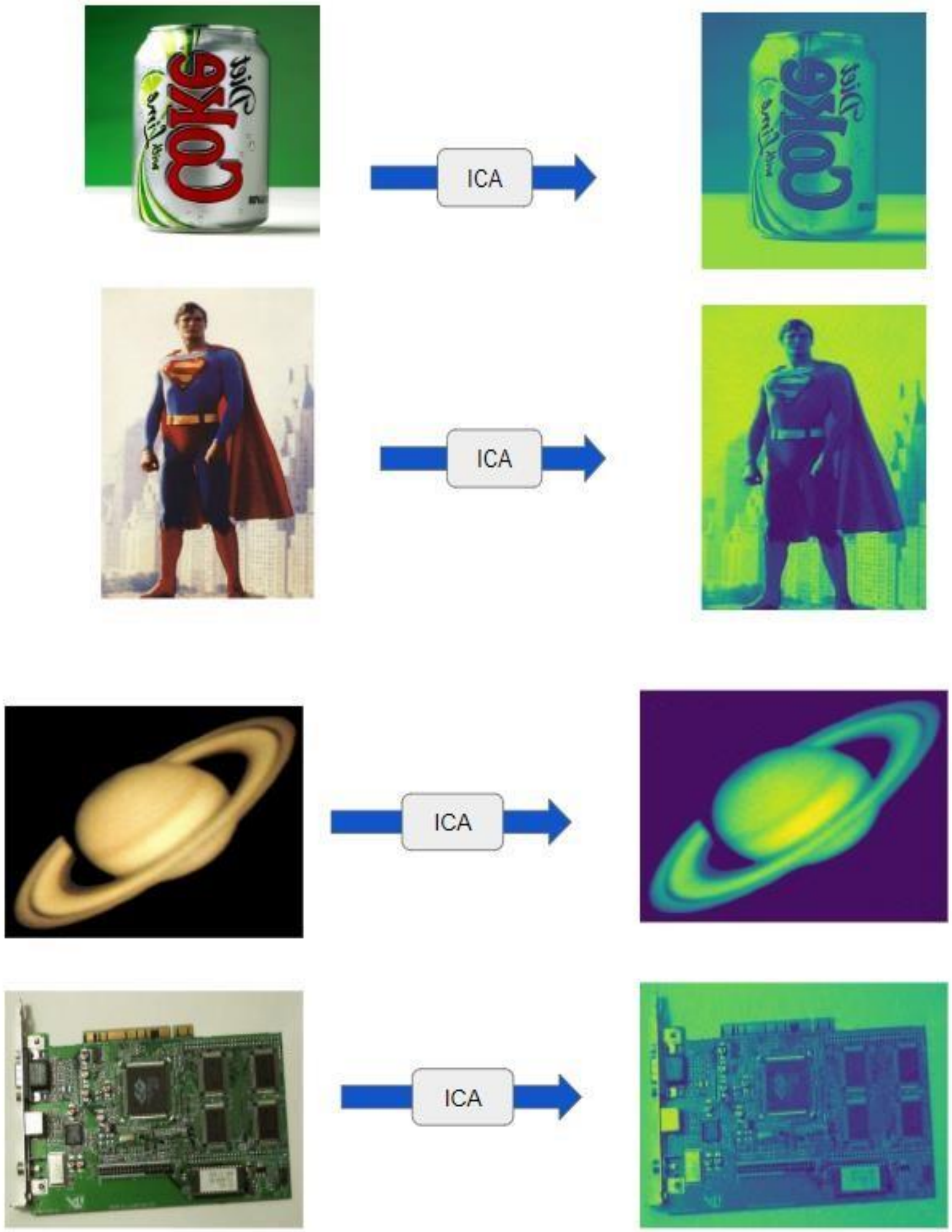
Fig. 4.2 PCA on the 4 classes from Caltech-256



4.2.2 Independent Component Analysis (ICA)

- Independent Component Analysis is a linear transformation technique in which the input is transformed into a new feature space where the features are statistically independent of each other.
- The mutual information between each of the output features is equal to zero, but the mutual information between input and output is maximized.
- In ICA, the variables are assumed to be a linear combination of some unknown latent variables and the combining system is also unknown.
- The latent variables are mutually independent, non-Gaussian and are said to be the independent components of the observed mixture.
- Unlike PCA the ICA vectors are not orthogonal to each other.
- All the components are equally important, unlike in PCA.
- Assume there exists N independent signals: $S = [s_1(t), s_2(t), \ldots, s_N(t)]$.
- . Let the linear combination of the signal be $Y(t) = A\ S(t)$
- A is knowns as the mixing matrix. A and S are unknown.
- The goal of the ICA is to recover original signals S(t) from Y(t), i.e., find the linear transformation ideally A^{-1} such that $A^{-1}\ Y(t) = S(t)$.
- We applied ICA with 25 and 50 components on the images from the 4 classes that we considered from Caltech-256.
- Inverse PCA is applied to get the feature extracted image

Fig. 4.3 ICA on the 4 classes from Caltech-256



4.2.3 Laplacian of Gaussian (LoG)

- Edges in an image are the areas of the image where there is a rapid change in the intensity of the pixels.
- Laplacian filters are used in finding the edges of an image because they are derivative filters.
- Since derivative filters are highly sensitive to noise, it is a common practice to carry out smoothening of the image using a Gaussian filter, ahead of the Laplacian.
- This process of smoothening the image and applying the Laplacian filter is called the Laplacian of Gaussian (LoG) operation.
- Laplacian filter is given by the equation:

$$L(x,y) = \nabla^2 f(x,y) = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2}$$

- The Gaussian filter is given by the equation

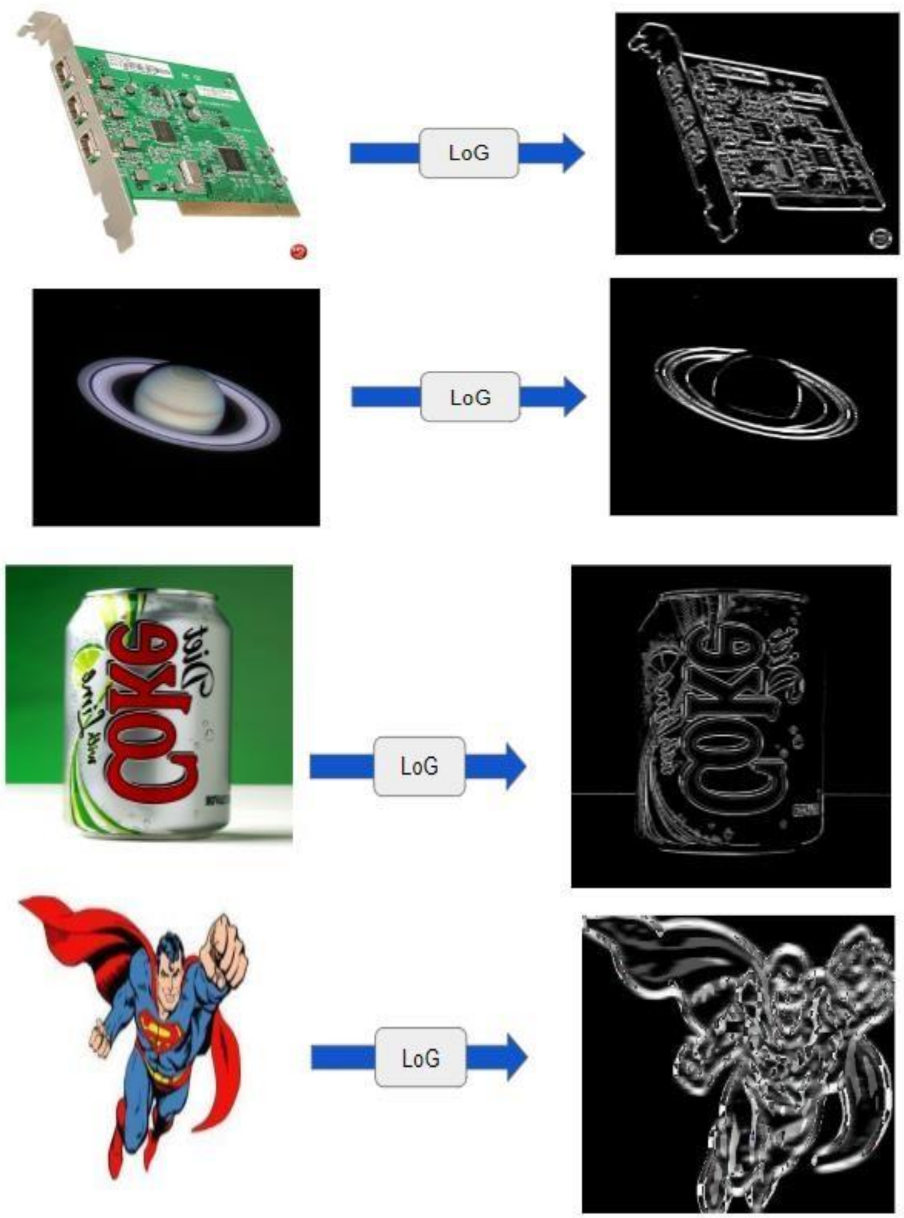
$$G_\sigma(x,y) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

- The Laplacian of Gaussian filter is therefore given by:

$$LoG(x,y) = -\frac{1}{\pi\sigma^4}[1 - \frac{x^2+y^2}{2\sigma^2}]e^{-\frac{x^2+y^2}{2\sigma^2}}$$

- We applied LoG where the Gaussian filter had **σ** = 3, on the images from the 4 classes that we considered from Caltech-256.

Fig. 4.4 LoG on the 4 classes from Caltech-256



4.2.4 Difference of Gaussian (DoG)

- DoG is another edge detection technique
- In the case of Difference of Gaussian, the image is first smoothened by convoluting it with a Gaussian kernel of width σ1.
- Using a different Gaussian kernel of width σ2, a different version of the smoothened image is obtained.
- The difference between these two Gaussian smoothened images, called the Difference of Gaussian, can be used to detect the edges in an image.
- Gaussian filter with variance σ, is given by :

$$G_\sigma(x,y) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

- The Difference of Gaussian filter is therefore given by,

  DoG = G(0,σ1) - G(0,σ2),

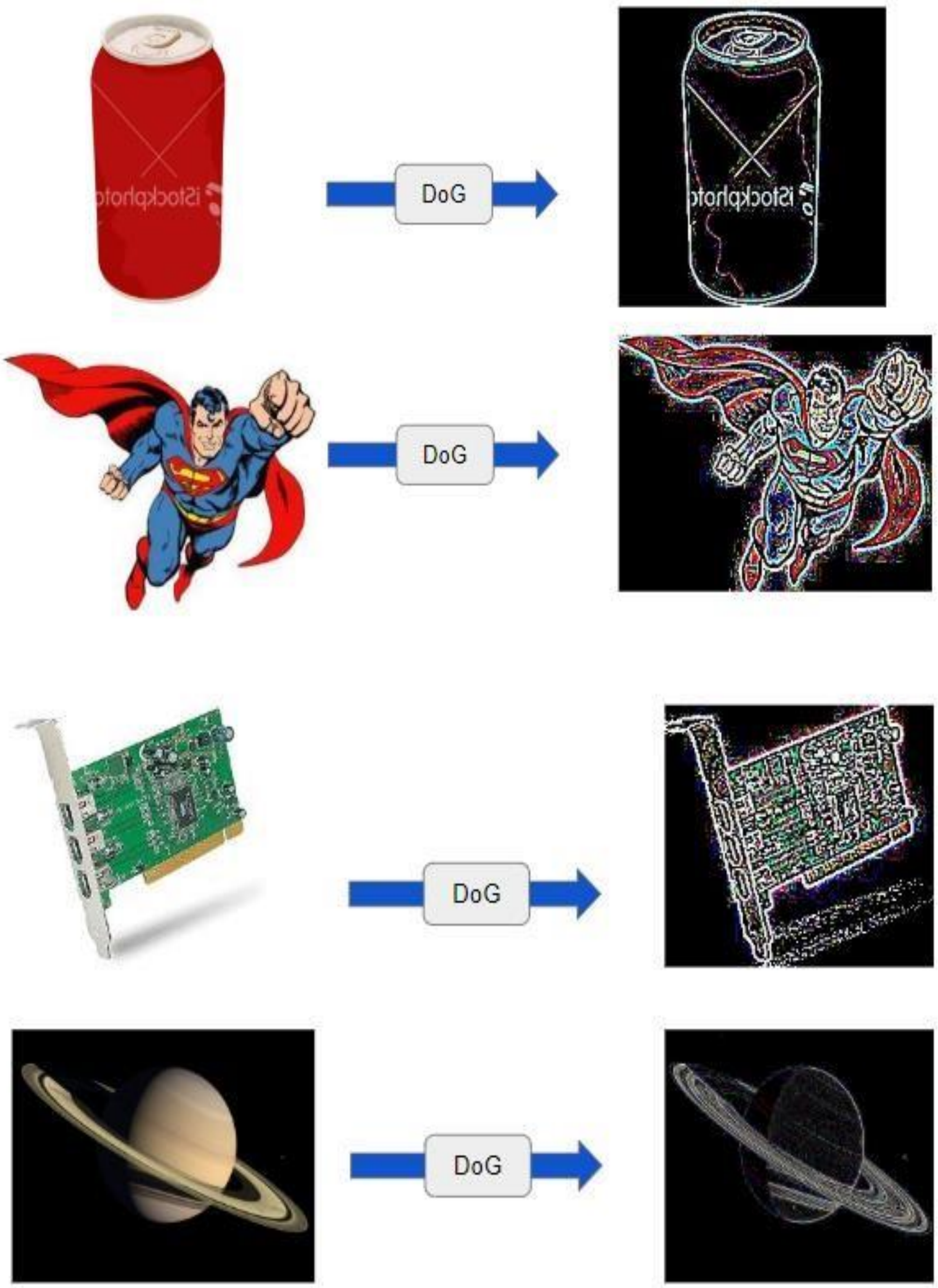
  where σ1 and σ2 are the variances of the two Gaussian filters considered.

- Thus mathematically, the output of DoG is given by :

  g(x,y) = DoG * f(x,y)

- We applied DoG with Gaussian filters **σ**1 = 3 and **σ**2 = 5, on the images from the 4 classes that we considered from Caltech-256.

Fig. 4.5 DoG on the 4 classes from Caltech-256



4.3 Convolutional Neural Network

Deep Learning is a part of Neural networks based machine learning. There are 3 methods of learning. They are supervised, unsupervised and semi-supervised. For our project, we have considered unsupervised learning. Traditionally machine learning techniques are linear, but deep neural networks are hierarchical and they are arranged in increasing order of complexity and abstraction.

When it comes to the state-of-the-art computer vision solutions, convolutional neural networks are at its core for a wide variety of such tasks. Since 2014 very deep convolutional networks became mainstream, mainly due to the increase in the processing capabilities of the machines, yielding highly acceptable and substantial gains across different problems. Although the increased model size and computational cost tend to translate to immediate quality gains for most tasks (as long as enough labeled data is provided for the training), computational efficiency and low parameter count are still enabling factors for various use cases such as mobile vision and bigdata scenarios.

Convolutional Neural Networks consists of neurons with tuneable weights and biases. Each neuron receives several inputs, takes a weighted sum over them, pass it through an activation function and responds with an output. The whole network has a loss function to calculate the error and update the weights and parameters. Unlike neural networks, where the input is a vector, here the input is a multi-channelled image (3 channelled in this case).

The CNN is made up of a number of different layers. The following are the typical layers found in a CNN:

- INPUT Layer:

  Holds the raw pixel values of the image, in this case, an image of width, height, and with three color channels R, G, B.

- CONVOLUTION Layer:

  Computes the output of neurons that are connected to local regions in the input, each computing a dot product between their weights and a small region they are connected to in the input volume.

- RELU Layer:

  Applies an elementwise activation function, such as the max(0,x) thresholding at zero. This leaves the size of the volume unchanged.

- POOL layer:

  Performs a down sampling operation along the spatial dimensions (width, height), resulting in volume.

- FC (i.e. fully-connected) layer:

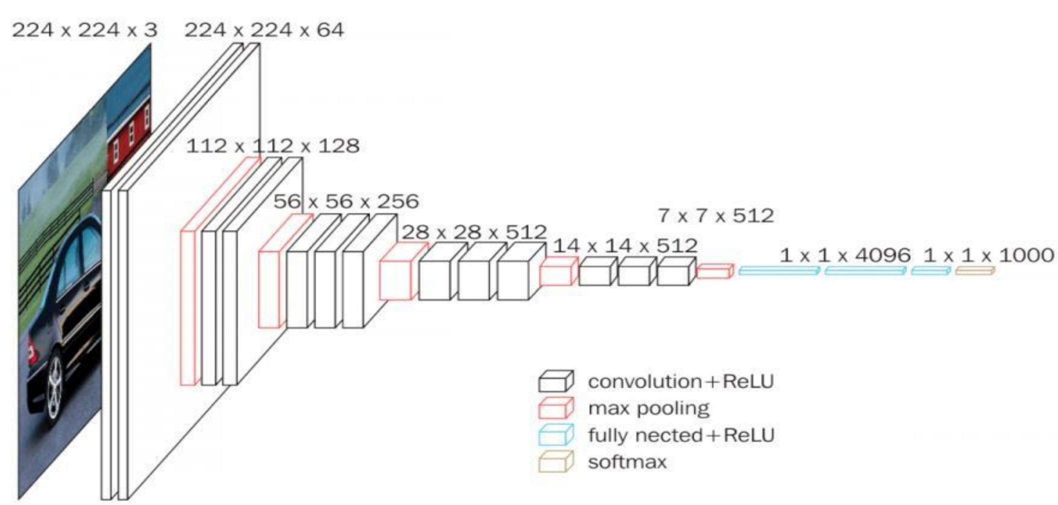
  Computes the class scores, resulting in a volume of size [1x1x No. Of classes]. As with ordinary Neural Networks and as the name implies, each neuron in this layer will be connected to all the numbers in the previous volume.

Since building a network from scratch is a tedious procedure, we used a pretrained network on our dataset. There are generally 2 ways of using a pretrained network on a new dataset, they are: transfer learning and fine-tuning. In case of transfer learning, the last fully connected layer of the pretrained network is removed and a new layer is added according to the new dataset for which the network should be trained. In the case of fine-tuning, the whole network is trained for some epochs to change the weights and learnable parameters according to the dataset it should be trained for.

We have considered 4 CNN Architectures pretrained on ImageNet dataset (it consists of 10,00 classes and over 14 million images). We have used transfer learning to train these pretrained networks on our dataset.

4.3.1 VGG-16 Architecture

Fig 4.6 VGG-16 Architecture

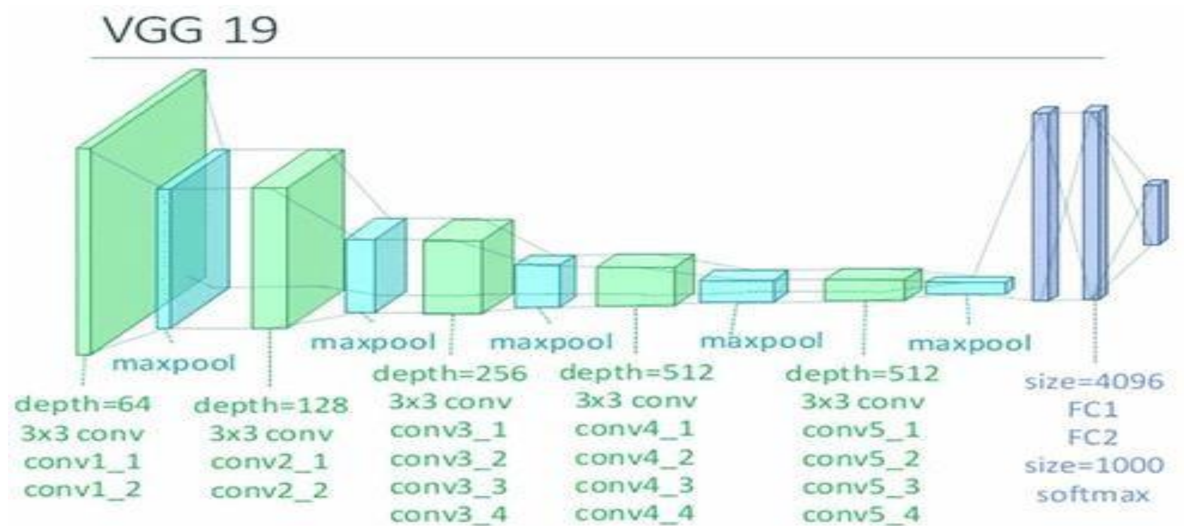


This architecture is from VGG group, Oxford. The VGG-16 network is characterized by 3x3 convolutional layers stacked on top of each other in increasing depth. Reducing the volume is handled by max-pooling. Two fully connected layers each with 4096 nodes, followed by another fully connected layer of 1000 nodes. Then this is followed by a soft-max classifier.

In VGG-16 the blocks are of the same filter size and are applied multiple times to extract more complex and representative features. This concept of blocks became common in the networks developed after VGG. It achieves the top-5 accuracy of 92.3 % on ImageNet which means the deep learning network is still improving by adding number of layers.

4.3.2 VGG-19 Architecture

Fig 4.7 VGG-19 Architecture

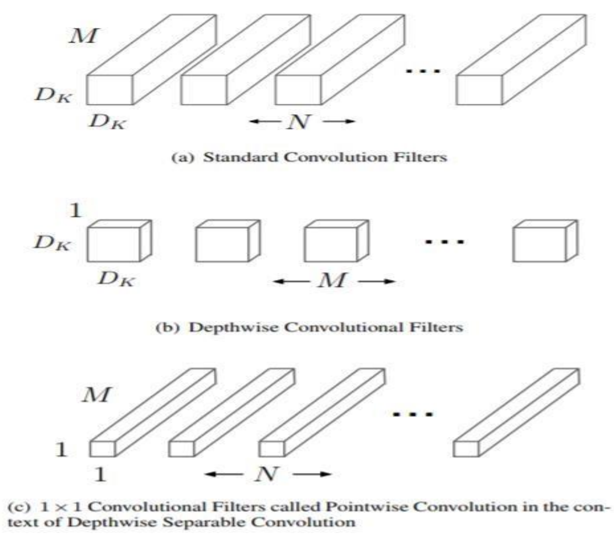


The VGG-19 Architecture is almost similar to that of VGG-16. It has 3 additional convolutional layers. VGG-19 obtains an accuracy of 91.0 % which means the deep learning network is NOT improving by adding number of layers. Hence, the authors of VGG Architecture stopped adding more layers to the network.

4.3.3 MobileNet Architecture

The MobileNet model is based on depth-wise separable convolution. Usually, standard convolution layers convolve over all channels of the input. In case of MobileNet, the convolution layers the standard convolution is factorized into a depthwise convolution and a 1x1 convolution called a point-wise convolution. The depthwise convolution applies a single filter to each input channel. The point-wise convolution then applies a 1x1 convolution to combine the outputs of the depth-wise convolution. This splitting of convolution into 2 layers makes it run much faster than other convolutional neural networks.

Fig 4.8 Building a depth-wise separable filter - The standard convolution filters in (a) are replaced by 3 layers: depth-wise convolution in (b) and point-wise convolution in (c)



There are 2 kinds of parameters that can be adjusted in the MobileNet architecture. They are width multiplier and resolution multiplier.

★ Width Multiplier :

Although the MobileNet architecture is small and has low latency, some applications may require the model to be smaller and faster. To construct these smaller models, they have introduced a parameter α, called width multiplier (α ∈ (0,1]). It's role is to thin a network uniformly at each layer. Then the number of input channels M becomes αM, and the number of output channels N becomes αN.

α = 1 is the baseline MobileNet and α<1 are reduced MobileNets.

★ Resolution Multiplier:

The 2nd hyper-parameter is the resolution multiplier ρ. This is applied to the input image, and the internal representation of every layer is reduced by ρ. (ρ ∈ (0,1])

ρ = 1 is the baseline MobileNet and ρ<1 are reduced MobileNets.

The input resolution of the network is 224,192,160 or 128.

Hence, MobileNet 1.0 224 implies that the input resolution is 224 and the α value is 1.

Fig 4.9 Depthwise Separable Convolution Block with depthwise and pointwise layers followed by BatchNorm and ReLu

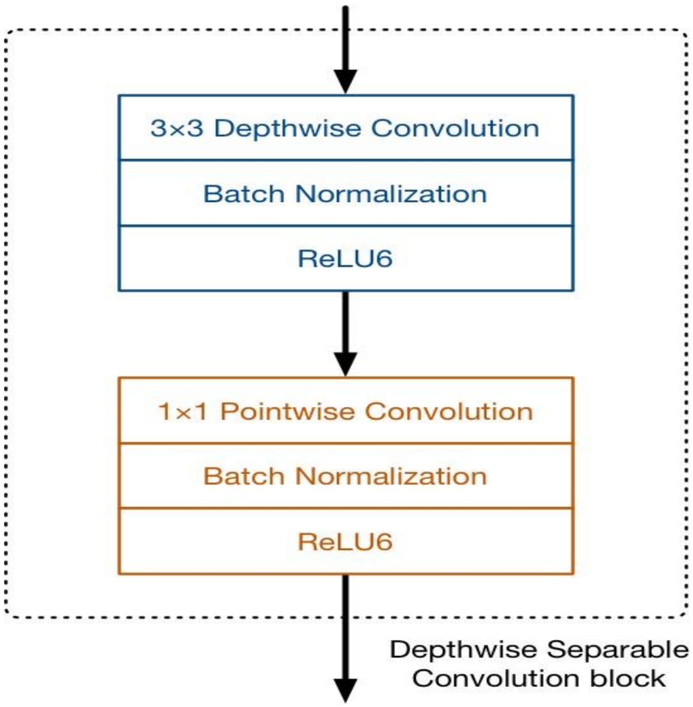


Table 4.1 MobileNet body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	3 × 3 × 3 × 32	224 × 224 × 3
Conv dw / s1	3 × 3 × 32 dw	112 × 112 × 32
Conv / s1	1 × 1 × 32 × 64	112 × 112 × 32
Conv dw / s2	3 × 3 × 64 dw	112 × 112 × 64
Conv / s1	1 × 1 × 64 × 128	56 × 56 × 64
Conv dw / s1	3 × 3 × 128 dw	56 × 56 × 128
Conv / s1	1 × 1 × 128 × 128	56 × 56 × 128
Conv dw / s2	3 × 3 × 128 dw	56 × 56 × 128
Conv / s1	1 × 1 × 128 × 256	28 × 28 × 128
Conv dw / s1	3 × 3 × 256 dw	28 × 28 × 256
Conv / s1	1 × 1 × 256 × 256	28 × 28 × 256
Conv dw / s2	3 × 3 × 256 dw	28 × 28 × 256
Conv / s1	1 × 1 × 256 × 512	14 × 14 × 256
5× Conv dw / s1	3 × 3 × 512 dw	14 × 14 × 512
5× Conv / s1	1 × 1 × 512 × 512	14 × 14 × 512
Conv dw / s2	3 × 3 × 512 dw	14 × 14 × 512
Conv / s1	1 × 1 × 512 × 1024	7 × 7 × 512
Conv dw / s2	3 × 3 × 1024 dw	7 × 7 × 1024
Conv / s1	1 × 1 × 1024 × 1024	7 × 7 × 1024
Avg Pool / s1	Pool 7 × 7	7 × 7 × 1024
FC / s1	1024 × 1000	1 × 1 × 1024
Softmax / s1	Classifier	1 × 1 × 1000

4.3.4 Inception v3 Architecture

Inception v3 is a widely used Neural network model. It has achieved an accuracy of around 78.1% on ImageNet Dataset. It is based on the original paper "Rethinking the inception architecture for computer vision" by Szegedy et. al.

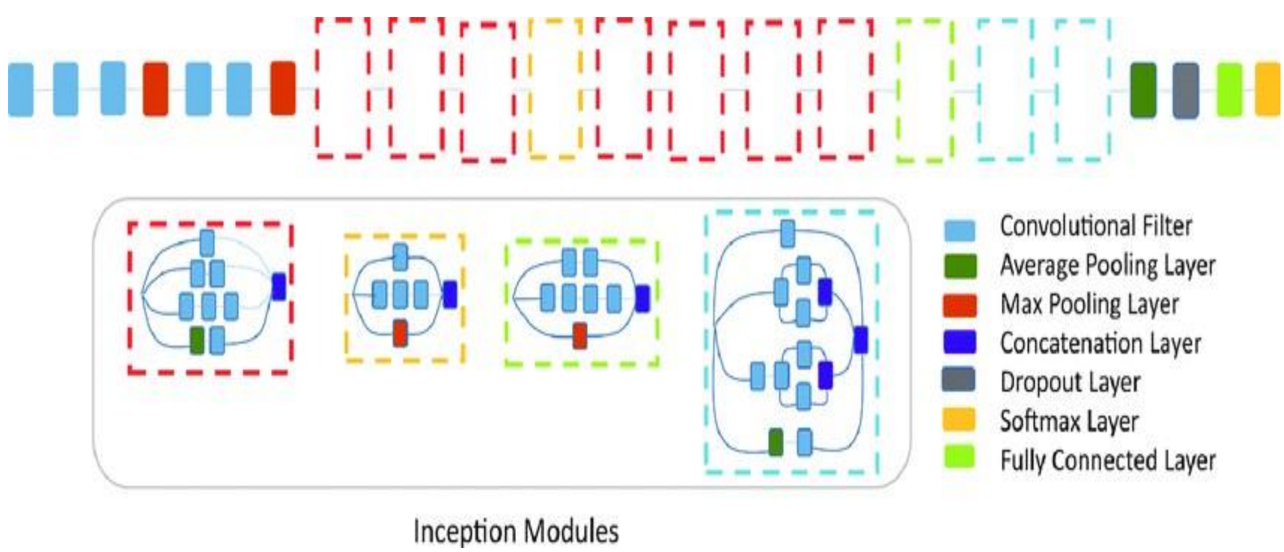
Inception has 4 versions. The 1st version was called GoogleNet which is now named as Inception v1. It uses the concept of factorizing convolutions to reduce the number of connections/parameters without decreasing the network efficiency. 2 3x3 convolutions replace 1 5x5 convolution. By using the above technique the number of parameters is reduced by 28%.

The architecture also uses factorization into asymmetric convolutions that is, 1 3x1 convolution followed by a 1x3 convolution replaces a 3x3 convolution. The number of parameters, in this case, is reduced by 33%.

Inception v3 uses 1 auxiliary classifier which is used as a regularizer.

Conventionally in architectures such as AlexNet and VGGNet the feature map downsizing is done by max-pooling, this is very expensive. In case of Inception v3 an efficient grid size reduction technique is used for downsizing. 320 feature maps are extracted by convolution with stride 2. 320 feature maps are also obtained by max pooling and these 2 sets of feature maps are concatenated as 640 feature maps which go to the next level of inception module.

Fig 4.10 Inception v3 Architecture



4.4 Results with Caltech-256

The input images first passed through one of the feature extraction methods (PCA, ICA, LoG, DoG, None). The outputs from these feature extraction methods were also images which were then passed through the CNN architectures mentioned above (refer Fig 3.1). For finding out the best network the above procedure was first applied on 4 classes of the Caltech-256 dataset namely PCI-card, Saturn, Soda-Can, and Superman.

PCA with two different number of output components was applied to the images. PCA with 25 extracted components was applied. The inverse of this PCA transformation gives back an image with those pixels contributing to the reduced components. This PCA with 25 components is abbreviated as PCA_{25}. Similarly, PCA was also attempted with 50 extracted components. This PCA with 50 components is abbreviated as PCA_{50}.

ICA was also attempted with 25 and 50 extracted components abbreviated as ICA_{25} and ICA_{50} respectively.

LoG was applied with a Gaussian filter of standard deviation 3.

DoG was applied with two Gaussian filters of standard deviations 3 and 5.

The results for the 4 class problem are depicted in the below table.

Table 4.2 Results of different classification techniques on 4 classes from Caltech-256

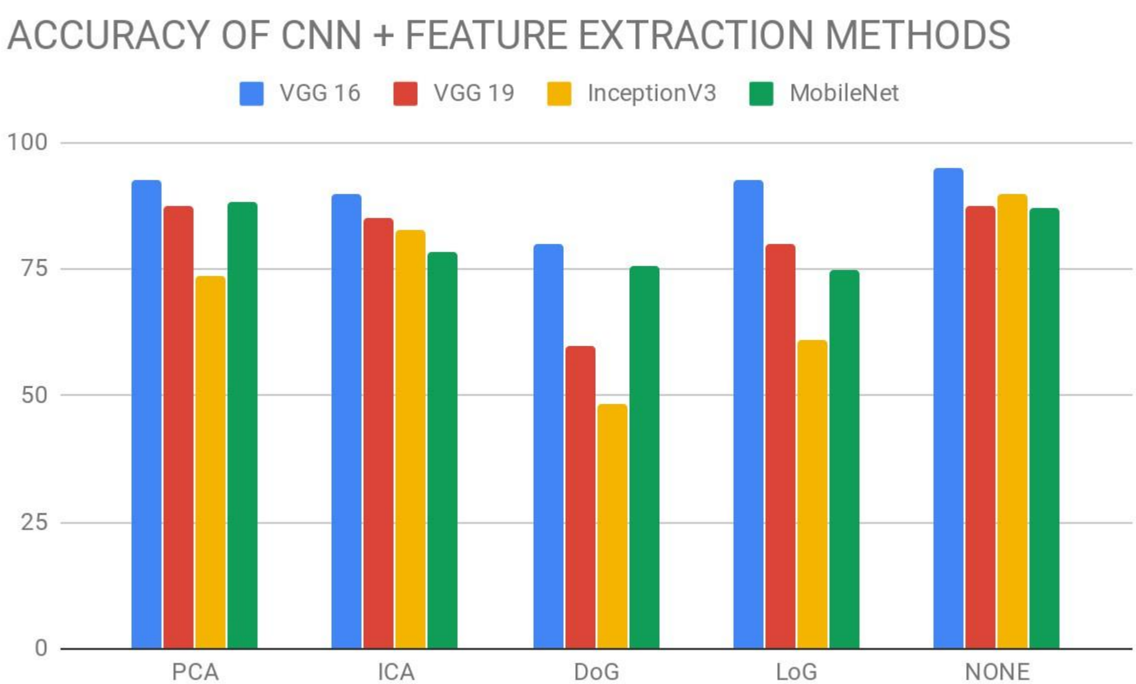
CNN Architecture	Number of Epochs	Feature Extraction Methods	Accuracy (%)
	20	-	**95**
		PCA_{25}	85
		PCA_{50}	87.5
		ICA_{25}	87.5

VGG – 16		ICA_{50}	90
		LoG	80
		DoG	50
	50	-	**92.5**
		PCA_{25}	90
		PCA_{50}	**92.5**
		ICA_{25}	87.5
		ICA_{50}	90
		LoG	**92.5**
		DoG	65
VGG - 19	20	-	90
		PCA_{25}	85
		PCA_{50}	87.5
		ICA_{25}	80
		ICA_{50}	90
		LoG	80
		DoG	60
		-	87.5
		PCA_{25}	85

	50	PCA_{50}	87.5
		ICA_{25}	80
		ICA_{50}	85
		LoG	80
		DoG	60
MobileNet	20	-	78.4
		PCA_{25}	82.1
		PCA_{50}	76
		ICA_{25}	78.4
		ICA_{50}	71.4
		LoG	61.1
		DoG	60
	50	-	87.2
		PCA_{25}	88.4
		PCA_{50}	70.6
		ICA_{25}	78.4
		ICA_{50}	71.4
		LoG	75
		DoG	75.8
	20	-	87.5
		PCA_{25}	73.5
		PCA_{50}	75.8
		ICA_{25}	80
		ICA_{50}	82.9
		LoG	60

Inception v3		DoG	48.5
	50	-	89.7
		PCA_{25}	72.1
		PCA_{50}	73.5
		ICA_{25}	81.1
		ICA_{50}	82.9
		LoG	61.1
		DoG	48.5

Fig 4.11 Graph showing results of different CNNs on 4 classes from Caltech-256.



It was found that the accuracies of the networks were far better when the images were input directly to the CNNs without applying any feature extraction methods. Out of the 4 networks, the best accuracy was given by VGG-16 without any feature extraction technique. This architecture was used for classification of our garbage dataset. Before training it on our garbage dataset, it was verified by checking it's accuracy for the entire Caltech-256 dataset. Using transfer learning the VGG-16 network achieved an accuracy of 50.3% over 50 epochs. This result is comparable with the results obtained by other researchers.

4.5 Results with our Garbage Dataset

The images from the garbage dataset were given as inputs to the 4 CNN networks that we had chosen. Since with the 4 classes of Caltech-256, the results were better without any feature extraction techniques, we chose to give the inputs directly without any feature extraction techniques. As described in Section 2.2, we have 3 categories of garbage, biodegradable, non-biodegradable and electronic waste and each of these categories have 2 classes. Biodegradable includes banana peel and paper cups, non-biodegradable includes crushed cans and plastic boxes, electronic waste includes circuit board and computer mouse.

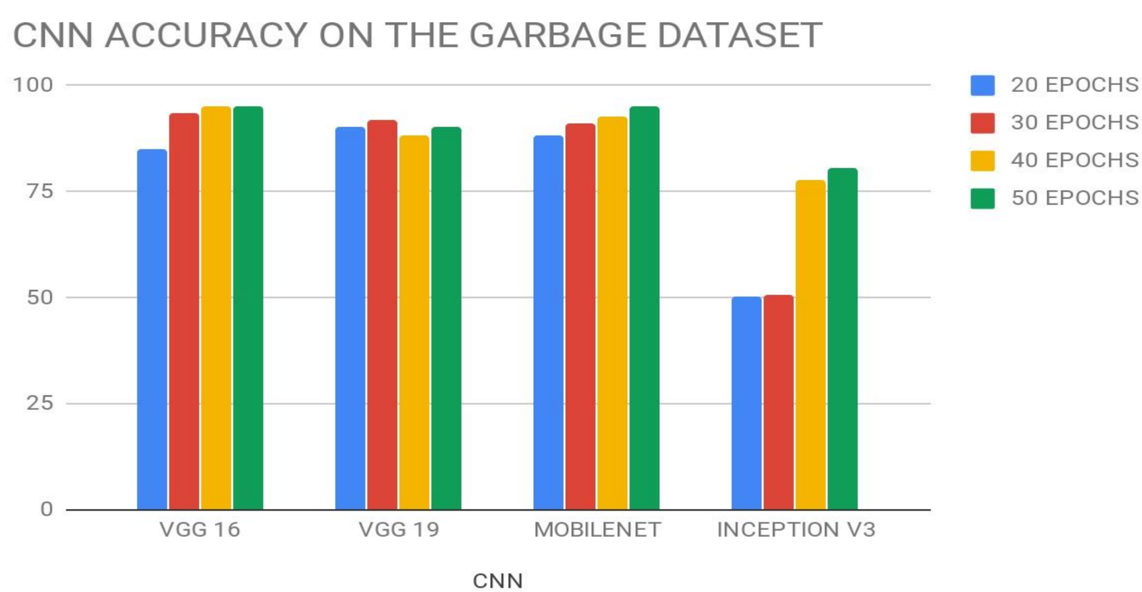
Table 4.3 Results of the 4 CNNs on our dataset.

CNN Architecture	Number of Epochs	Accuracy (%)
VGG-16	20	85.17
	30	92.52
	40	95.19
	50	95.19
VGG-19	20	90.18
	30	91.85
	40	88.19
	50	90.18
MobileNet	20	88.1
	30	91

	40	92.5
	50	95
	20	50.3
Inception v3	30	50.7
	40	77.6
	50	80.6

The results are shown in graphical form in the below figure:

Fig 4.12 Graph showing results of different CNNs on our dataset.



The VGG-16 architecture showed the best results and converged faster for our garbage dataset. Therefore, this architecture was chosen for classifying the garbage in real-time.

5. Hardware

5.1 Need for Hardware

However satisfying the accuracies of the CNN models might seem, it can be used in the real world only if it's integrated with some sort of hardware. Also, the visual proof or the "live demo" is much more convincing than the software accuracies.

Hence, we propose a simple stationary base and a claw mechanism, integrated along with a camera so as to let the model work in real time.

5.2 Mechanism

The object is placed at the position labeled 'x' which is in front of the camera. The camera clicks the photo of the garbage placed in front of it, feeds it as the input to the CNN classifier. The CNN classifier predicts the object and classifies it into one of the 3 categories of wastes which the problem is trying to solve. Upon classification, the claw picks up that object and drops it at the right position. Upon successful completion of the process, the next object is placed and so on…

Upon doing the same for all the entire garbage, the wastes would be segregated into biodegradable, non-biodegradable and e-waste each of which would be dropped at a different location.

5.3 Building the hardware

The subsequent sections give a detailed explanation of the construction of the hardware. It mainly focuses on the materials required and construction.

5.3.1 Materials Required

The following materials are required for the construction of the hardware:

- Base:

  A heavy base is required upon which rests the entire structure. It is essential that the base is heavy so as to ensure that the structure above it remains stable. The base we have used in the experiment is a heavy wooden block.

- A secondary base:

  A secondary lighter base is required which can be rotated so as ensure that the object picked up is dropped at different locations based on its category.

- Claw :

  A claw is required to pick up the objects. This claw is mounted on a rod which is in turn fixed on the secondary base

- TowerPro MG996R servo motor:

  MG996R is a 180-degree high torque servo (11 kg.cm) which is used to rotate the secondary base.

- TowerPro MG995 servo motor:

  MG995 is also a high torque 360-degree servo motor. It is used in the opening and closing of the claw.

- Few tools like a hammer, nails, bolt and nuts, screwdrivers and clamps.
- External power supply to power the motors. ● Raspberry Pi:

  The raspberry pi is a processor. The weights from the trained network are copied to the raspberry pi. These weights are then used to predict the garbage picked.

Fig 5.1 (a) TowerPro MG996R servo motor (b) TowerPro MG995 servo motor

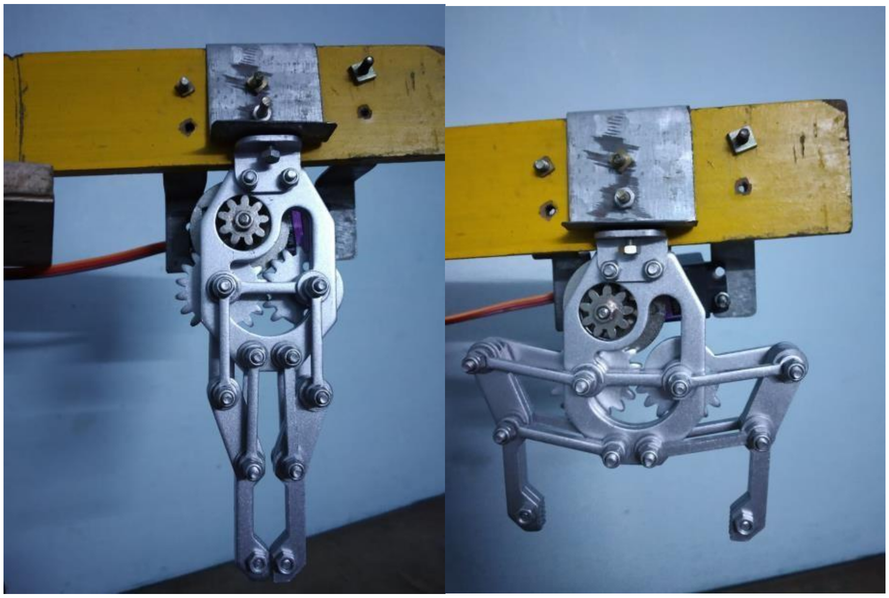
(a)

(b)

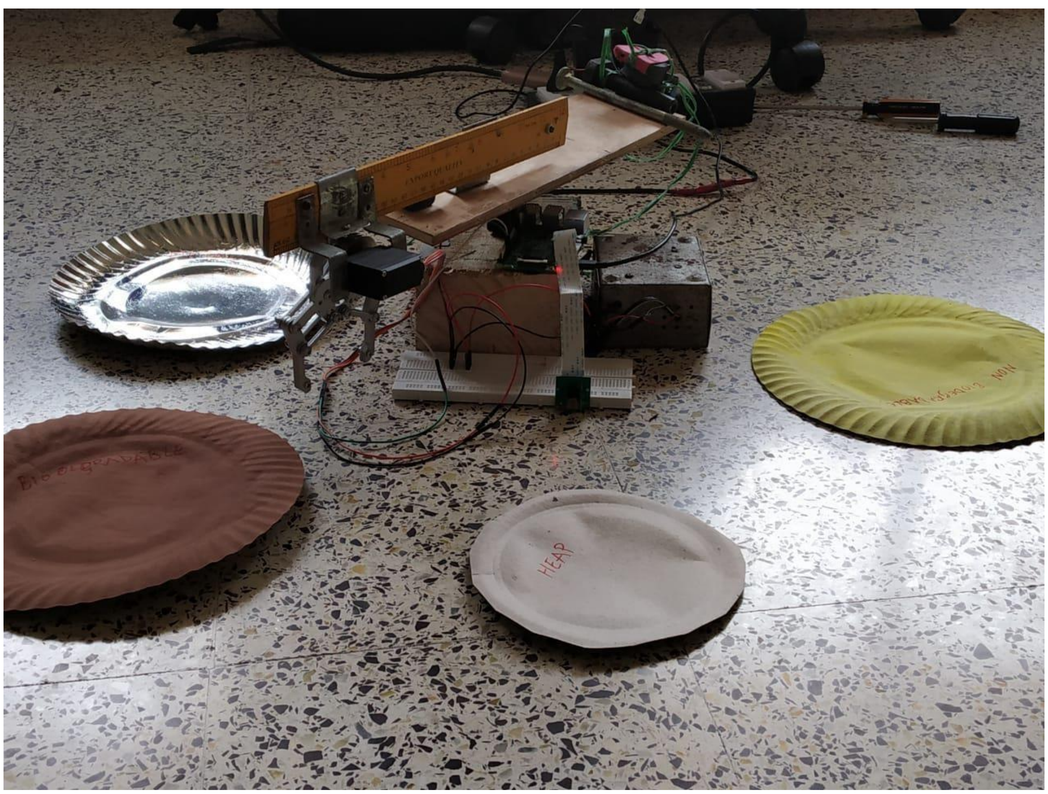
5.3.2 Construction

- Initially, fix the MG996R servo on the wooden base. It is fixed using nails and hammering them into the wood. Care should be taken so as to not damage the motor while hammering the nails in and at the same time the motor should be fixed firmly onto the base.
- Fix the rod on the secondary base using an L-clamp.
- Now attach the claw onto the rod.
- Fix the MG995 motor in the place near the claw using clamps.

Fig 5.2 a)The claw



b) The Hardware setup, which includes, the crane and the bins



c) Biodegradable bin

d) Non-Biodegradable Bin

e)Electronic waste Bin **f) Heap**

5.3.3 Getting the motors ready

The following sequence of events is carried out :

- Initially, the base motor is rotated by 90 degrees which enable the claw to reach right above the object
- Now the motor attached to the claw is triggered and thus the claw closes and picks up the object
- Based on the category of the waste, the base motor again rotates by a certain angle which makes the claw reach the point where the object is supposed to be dropped
- The claw motor is triggered again in the opposite direction and this drops the object.

6. Integrating Software and Hardware

6.1 Introduction to Raspberry Pi

Now that the hardware is constructed, it's time we put into use along with the software. But before that, we need to run the software in real time. We can do so using Raspberry pi.

Raspberry Pi is a small cchip-basedmicrocontroller or a computer that can perform various tasks ranging from simple mathematical operations to image recognition, classification , and other heavy Machine Learning algorithms. We installed Raspbian Operating System on the Rpi. In our application, we are using rthe aspberry pi to run our Convolutional Neural Network classifier in combination with a few other resources. The other resources are mainly tensorflow and python. Using tensorflow and RPI applications like object detection and classification is brought about in ease.

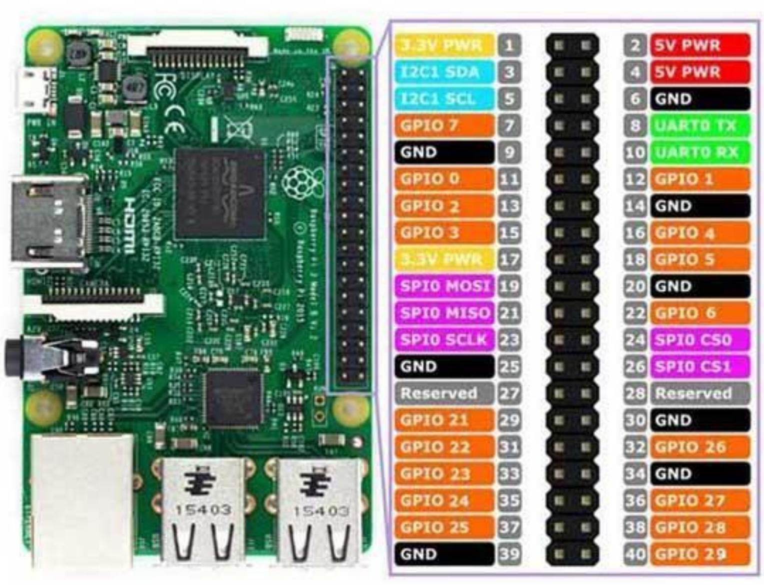
To run the network on the Rpi, we first need to check if the correct libraries are installed. In our case, we needed tensorflow, scipy, matplotlib, Keras open cv2, numpy.

- Tensorflow is an open source Machine Learning library which consists of various applications for speech detection, video detection, text detection, object detection. Tensorflow has different sub-librariesin which are application specific. Keras and tensorflow together helps us to create neural network line by line. Using keras tensorflow, building a deep network is user friendly. We used keras to build the network / model, loading the images, converting the images to array, etc.
- Scipy is also a free and open source software used for various computing techniques. Scipy contains various modules such as Linear algebra, Fast Fourier Transform, optimization, Image processing.
- Matplotlib is a python plotting tool to plot various barplots, histograms, power spectra, etc.
- Open cv is a python library that includes several hundred computer vision algorithms. It consists of the Image processing, Object Detection, Video Analysis, etc. modules. We used open cv to read the image from the given path and also to label the images.

Open cv2 has automatic memory management. So efficient use of memory is ensured.

- Numpy is another python library for scientific computing. It is a general purpose array-processing package. We use numpy for various operations like saving the weights in a .npy file, to load the .npy files, etc.

Fig 6.1 Raspberry Pi model 3B



As we notice, the Raspberry pi has 40 pins. Each pin has a specific function. It also has 2 CSI interfaces, 4 USB ports, 1 ethernet cable, SD card holder under the board, a port to connect to power supply. We use the pins which support PWM for the motors. The motors use pins which provide duty cycles.

6.2 Introduction to Raspberry Pi camera

The Rpi Camera module is a custom design add on for the RPi. The Rpi has an interfacing socket to which the ribbon cable can be easily attached. This interface uses the dedicated CSI interface, which was designed especially for interfacing to cameras. The CSI bus is capable of extremely high data rates, and it exclusively carries pixel data.

Raspberry pi camera module consists of 2 parts. The first part consists of the board on which there is a camera and the Sunny IC. The second part consists of the flexible ribbon cable. The board is around 3gms and the camera is an 8 pixel camera.

1. Update RPi firmware and raspi-config to the camera by using the commands ‘apt-get update’ or ‘apt-get upgrade’.
2. Go to Raspi-config, go to ‘interfaces’, then ‘camera’, click on Yes if you want to interface the camera. Then reboot the system.
3. Using the command-line type raspivid and raspistill operate a camera to capture video clips or images respectively. Command : raspistill -o Image.jpg

Fig 6.2 Raspberry Pi camera module

6.3 Simultaneous working of hardware and software

The weights from the trained network are saved in a .npy file. These files are then copied on to the Raspberry pi. These weights are then accessed using numpy in another code. The code consists of the last layer and then the predict function, which is used to predict the object and it's category.

Upon getting the software to work in real time, the following methodology is followed:

As we see from the flow chart, upon the start of the process, the raspberry pi send a trigger signal to picamera to click a photo of the object. The clicked object is passed through the CNN classifier which predicts the category of waste it belongs to. This result is stored in the memory of the raspberry pi which will be used later.

Now, the claw has to move from its initial position to reach the place where the object is kept. This motion is brought about by the movement of the base motor by 90 degrees. Now the claw motor rotates 180 degree to bring about the complete closure of the claw. This motion helps to hold the object tight while moving it to its right position.

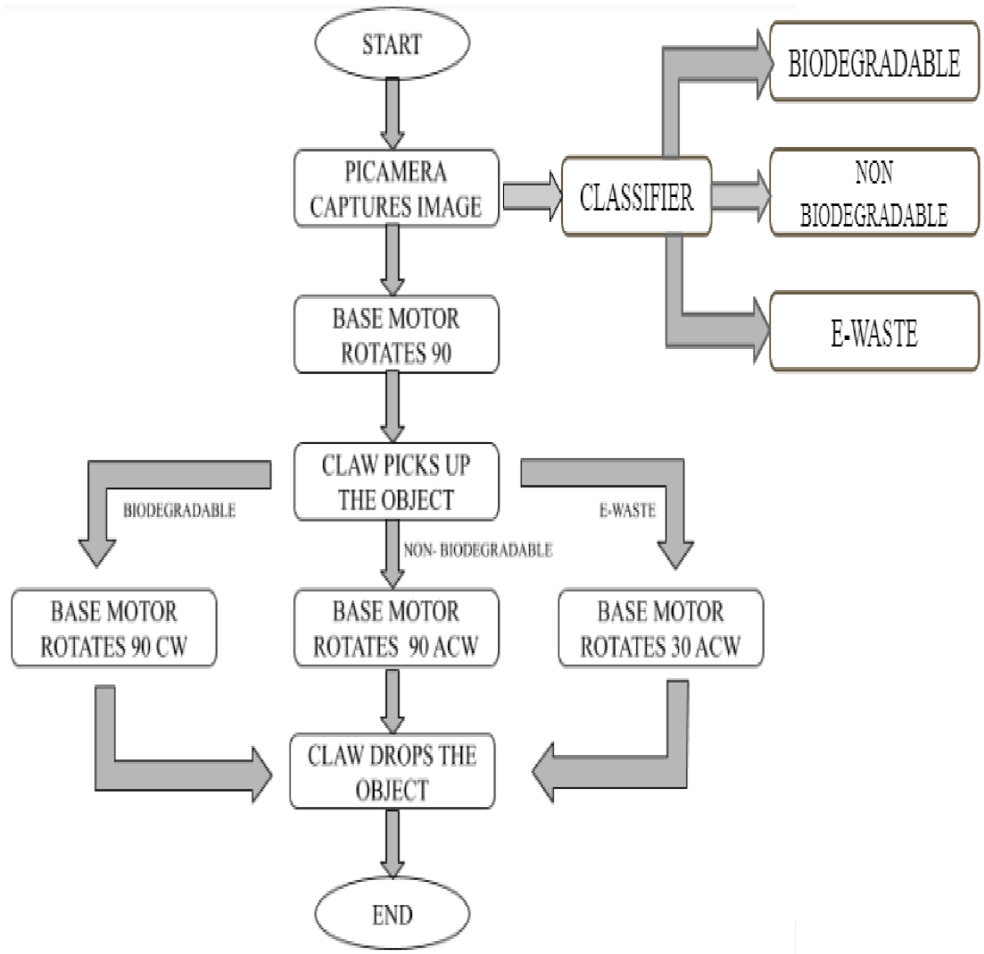
Now upon picking up the object, it's of utmost importance that we drop it in the right place. This is when the CNN classifier's predicted results comes in handy. Based on its prediction, we move the base motor by a certain angle which determines its drop location. Thus, different predictions rotate the base motor by different angle which implies that the waste is dropped at different locations.

Now, when the claw reaches the right position , the claw motor is rotated 180 degrees in the opposite direction and this results in dropping the object.

This series of steps ends the process.

The same is repeated for the next object placed and the cycle continues.

Fig 6.3 Flowchart showing simultaneous working of hardware and software



7. Results

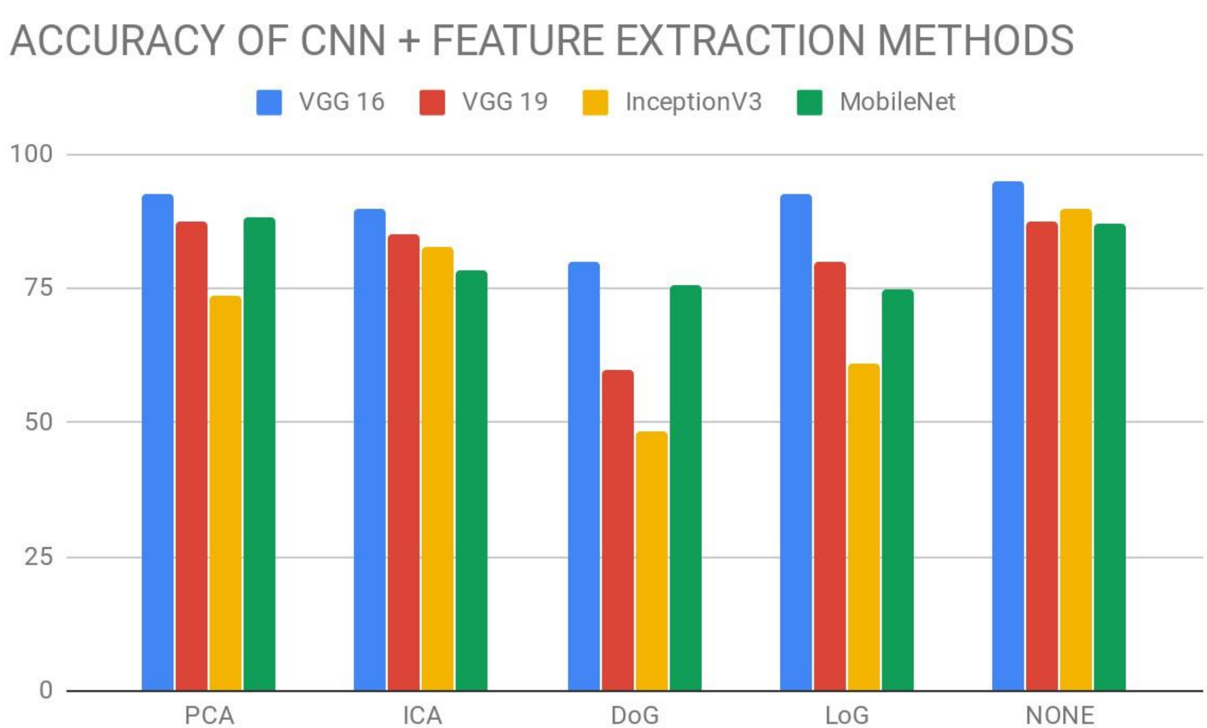
As we come to the end of the project, it is time we tabulate our results.

7.1 Choice of CNN and feature extraction method on the Caltech 256

As mentioned earlier, multiple CNNs with multiple feature extraction methods were tried. It was observed that the CNNs performed the best when no feature extraction methods were used. Among the CNNs, VGG16 outperformed all the other CNNs. The maximum accuracy obtained was 95% for VGG16 without any feature extraction.

The graph justifies the above statement.

Fig 7.1 Graph of accuracies of combination of various feature extraction methods and CNNs

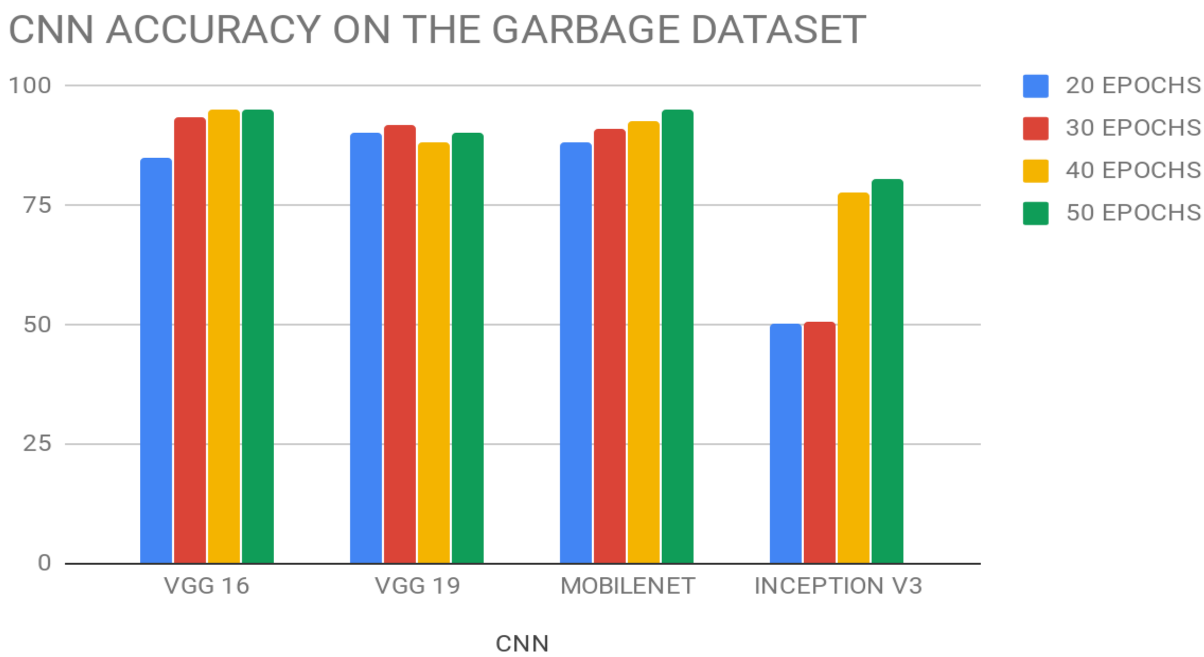


7.2 Choice of CNN on the garbage dataset

In the above subsection we realised that the VGG16 performs best on the caltech 256 dataset. Just for further confirmation, all the CNNs were tried on the garbage dataset as well. Unsurprisingly, VGG16 was better than the other CNNs as shown in the graph.

The maximum accuracy was obtained for VGG16 and MobileNet at 95%.

Fig 7.2 Accuracies of various CNNs on Garbage Dataset



7.3 Real time prediction

It’s important that we also implement the model in real time as well. As mentioned earlier, it was carried out by clicking photo of the object by the pi-camera and passing the raw image through the classifier. Few predictions are as shown :

Fig 7.3 Real time predictions

8. FUTURE SCOPE

- Though the VGG-16 architecture gives very high accuracy with still images, it doesn’t perform so well on real time images captured from the pi-camera. This could be improved by removing the background dynamically and replacing it with a uniform background before passing the image from pi-camera to the neural network.
- Efficient cropping techniques could also be used on the image from pi-camera before passing it through neural network, so that the image obtained, focuses on the object.
- Transfer learning does not extract the features of each class very accurately, since the weights have been pretrained on ImageNet dataset. Only the last few layers are trained according to the actual dataset to be trained. Therefore, a better neural network could be built from scratch without using transfer learning.
- The hardware structure can be improved by adding a conveyor belt for positioning the waste at the location where the claw picks it up at regular intervals and later drops it in corresponding location based on the prediction.
- The hardware could also be improved by making the crane mobile, and thus enabling it to move towards the garbage, instead of bringing the garbage to the claw.
- Though various positions of the objects have been taken into consideration with different backgrounds, the dataset could be improved by adding more images to each class.

Bibliography

Although a lot of our efforts have gone into the successful completion of the project, this wouldn't have been possible without a vast amount of knowledge and ideas we received from various research papers, blogs and its implementation. The research papers and blogs are listed below:

[1]Deep Residual Learning for Image Recognition He et al.2015

[2]Fitzwatler G. Ang1, Maria Karla Angel R. Gabriel, "Automated waste segregator with mobile robot delivery waste system", Department of Electronics and Communications Engineering De La Salle University, Manila

[3]Aleena V.J.*, Kavya Balakrishnan, "Automatic Waste Segregator and Monitoring System", Department of Electronics and Communication Engineering, Holy Grace Academy of Engineering, Thrissur, Kerala, India

[4]Amrutha Chandramohan, Joyal Mendonca,, "Automated Waste Segregator", Rashtreeya Vidyalaya College of Engineering (R.V.C.E.)

[5]Aguda, John Darell C. Bondad, Mariah Rayah D., "Design and implementation smart-solar powered automatic waste segregating machine with image processing and monitoring"

[6]Rashmi M. Kittali and Ashok Sutagundar, " Automation of Waste Segregation System using PLC ", Basaveshwar Engineering College, Bagalkot, India

[7]Alicia L. Castillo et al., "Status of Solid Waste Management in the Philippines", International Resource, University of Kitakyushu, Japan, 2013.

[8]Daniel Hoornweg et al., "WHAT A WASTE A Global Review of Solid Waste Management", Urban Development & Local Government Unit World Bank, Washington, DC., No.15, Mar. 2013.

[9]M.K. Pushpa, et al., "Microcontroller based Automatic Waste Segregator." International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering. Vol. 3, Issue 5. May 2015.

[10] M Kayva et al., "Sensor Based Smart Dustbin for Waste Segregation and Status Alert", College of Engineering, Rajarajeswari College of Engineering, India, 2017.

[11] Elfasakhany A. et al., "Design and Development of an Autonomous Trash Sorting System", Department of Mechanical Engineering, Taif University, Saudi Arabia, 2015.

[12] Minal Patil et al., “Implementation of Automated Waste Segregator at Household Level”, Department of Electronics Engineering, V.E.S.I.T, Mumbai, India, 2017

[13] S.M. Samreen et al., “Automatic Metal, Glass and Plastic Waste Sorter “, International Journal for Research in Applied Science & Engineering Technology (IJRASET), 2017.

[14] A.T. Garcia et al., “Intelligent Waste Separator”, Department of Electronics, Systems and IT, Mexico, 2015

[15] Kihyuk Sohn, Dae Yon Jung, “Efficient Learning of Sparse, Distributed, Convolutional Feature Representations for Object Recognition”, Dept. of Electrical Engineering and Computer Science University of Michigan, Ann Arbor, MI 48109

[16] Kaiming He, Xiangyu Zhang, “Deep Residual Learning for Image Recognition”, Microsoft Research

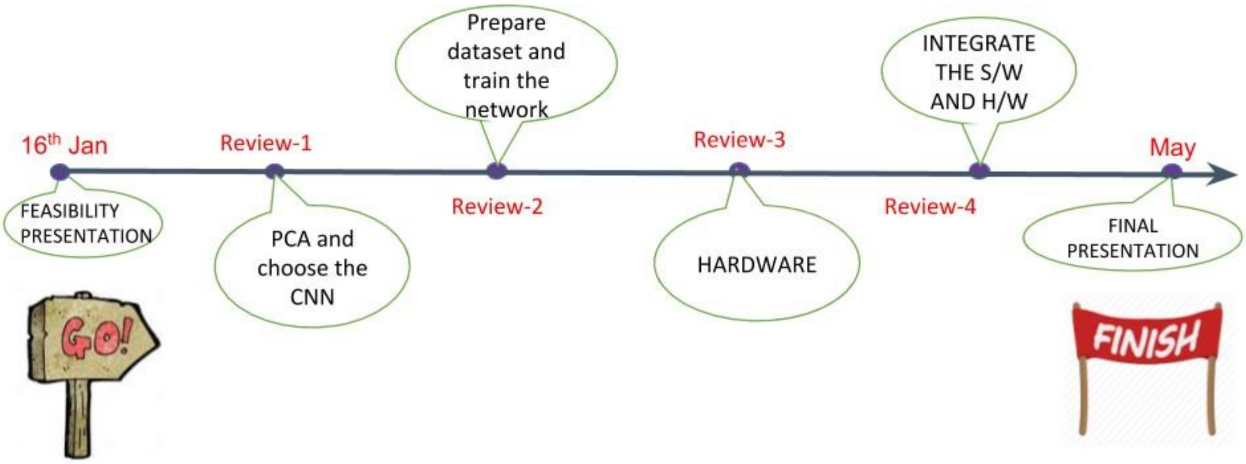
[17] Bajaj JS., Urban Solid Waste Management in India, New Delhi: Planning Commission Government of India; 1995.

[18] Daniel Hoornweg, et al. What a Waste: A Global Review of Solid Waste Management. Washington, DC: Urban Development & Local Government Unit World Bank, No.15; Mar 2012.

[19] Priya B. K., T. Lavanya, V. Samyukta Reddy, Yarlagadda Pravallika, “Bin That Think’s”, The International Journal Of Science & Technoledge.

[20] Andrew G. Howard, et al. , MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, Google Inc.

Appendix A: Project Timeline



Review-1 was held on 15th February 2019,

Review-2 was held on 15th March 2019,

Review-3 was held on 5th April 2019,

Review-4 was held on 26th April 2019.

Appendix B

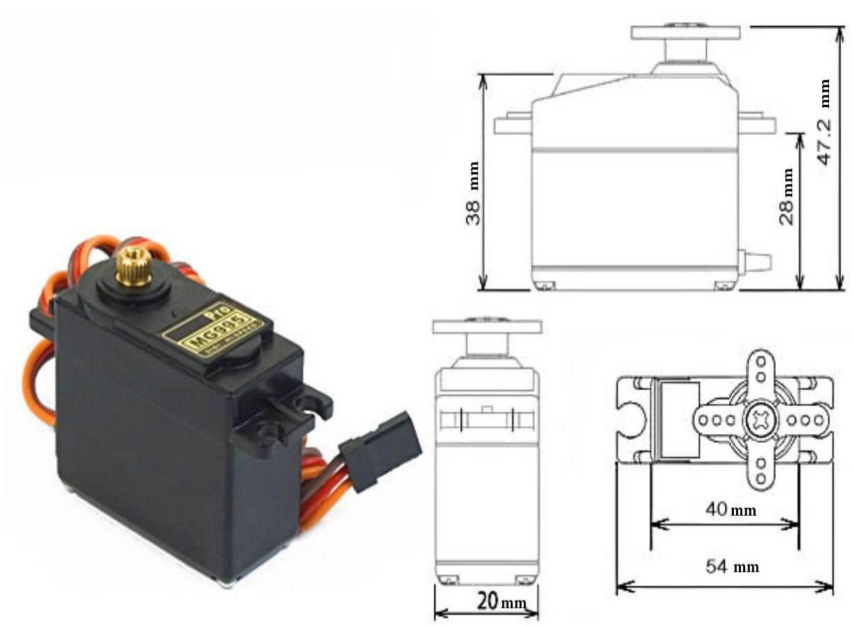
31150-MP
MG995 High Speed Servo Actuator

The unit comes complete with color coded 30cm wire leads with a 3 X 1 pin 0.1" Pitch type female header connector that matches most receivers, including Futaba, JR, GWS, Cirrus, Blue Bird, Blue Arrow, Corona, Berg, Spektrum and Hitec.

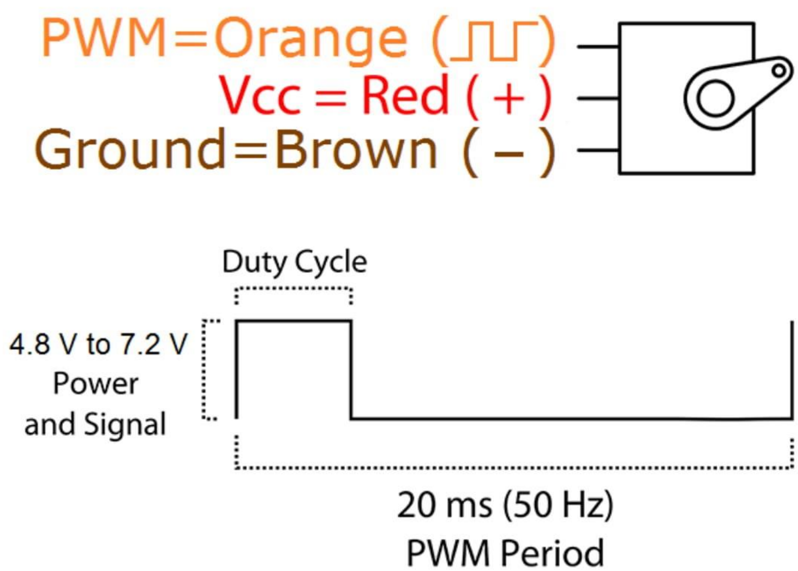
This high-speed servo actuator is not code dependant; You can use any servo code, hardware or library to control them. The MG995 Actuator includes arms and hardware to get started.

Specifications

- Weight: 55 g
- Dimension: 40.7 x 19.7 x 42.9 mm approx.
- Stall torque: 8.5 kgf·cm (4.8 V), 10 kgf·cm (6 V)
- Rotation Angle: 120deg. (+- 60 from center)
- Operating speed: 0.2 s/60º (4.8 V), 0.16 s/60º (6 V)
- Operating voltage: 4.8 V to 7.2 V
- Dead band width: 5 µs
- Stable and shock proof double ball bearing design
- Metal Gears for longer life
- Temperature range: 0 ºC – 55 ºC



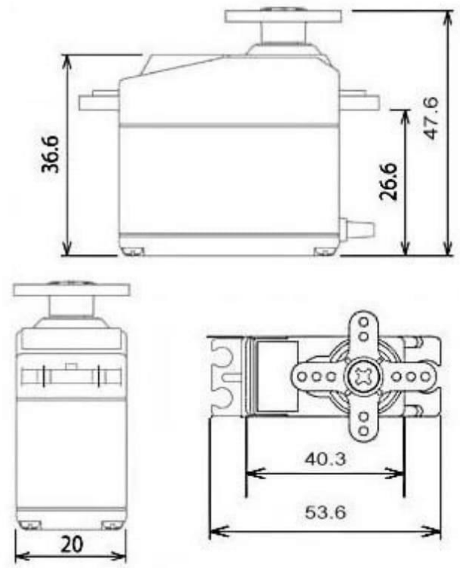
31150-MP
MG995 High Speed Servo Actuator



Information obtained from or supplied by mpja.com or Marlin P. Jones and Associates inc. is supplied as a service to our customers and accuracy is not guaranteed nor is it definitive of any particular part or manufacturer. Use of information and suitability for any application is at users own discretion and user assumes all risk.

MARLIN P. JONES & ASSOC., INC.
P.O. Box 530400 Lake Park, Fl 33403
800-652-6733 FAX 561-844-8764
WWW.MPJA.COM

MG996R High Torque
Metal Gear Dual Ball Bearing Servo



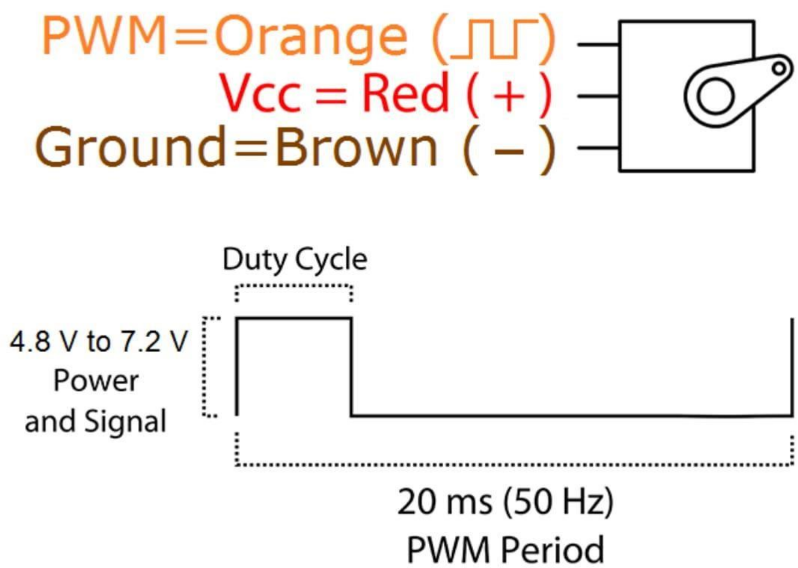
This High-Torque MG996R Digital Servo features metal gearing resulting in extra high 10kg stalling torque in a tiny package. The MG996R is essentially an upgraded version of the famous MG995 servo, and features upgraded shock-proofing and a redesigned PCB and IC control system that make it much more accurate than its predecessor. The gearing and motor have also been upgraded to improve dead bandwith and centering. The unit comes complete with 30cm wire and 3 pin 'S' type female header connector that fits most receivers, including Futaba, JR, GWS, Cirrus, Blue Bird, Blue Arrow, Corona, Berg, Spektrum and Hitec.

This high-torque standard servo can rotate approximately 120 degrees (60 in each direction). You can use any servo code, hardware or library to control these servos, so it's great for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. The MG996R Metal Gear Servo also comes with a selection of arms and hardware to get you set up nice and fast!

Specifications

- Weight: 55 g
- Dimension: 40.7 x 19.7 x 42.9 mm approx.
- Stall torque: 9.4 kgf·cm (4.8 V), 11 kgf·cm (6 V)
- Operating speed: 0.17 s/60º (4.8 V), 0.14 s/60º (6 V)

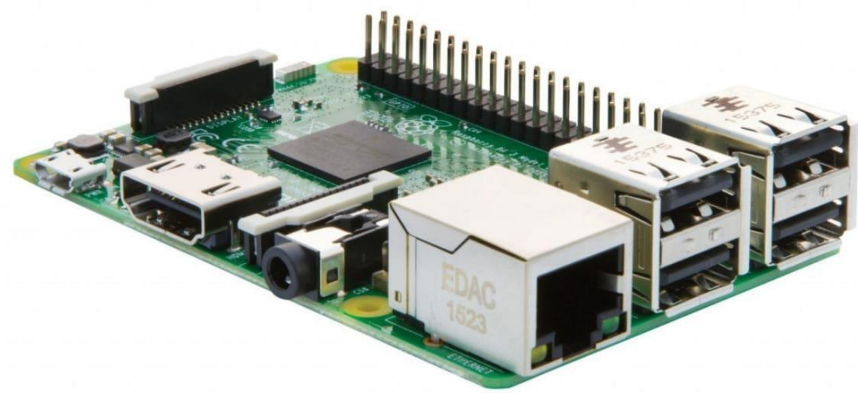
- Operating voltage: 4.8 V a 7.2 V
- Running Current 500 mA – 900 mA (6V)
- Stall Current 2.5 A (6V)
- Dead band width: 5 µs
- Stable and shock proof double ball bearing design
- Temperature range: 0 ºC – 55 ºC

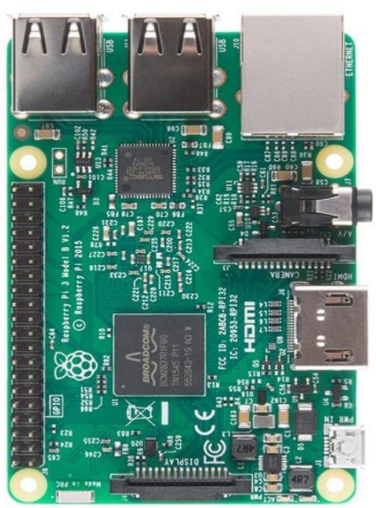


Farnell element14 Newark element14

http://uk.farnell.com/buy-raspberry-pi http://www.newark.com/buy-raspberry-pi

RASPBERRY PI 3 MODEL B





Product Name: RASPBERRYPI3-MODB-1GB

http://uk.farnell.com/buy-raspberry-pi http://www.newark.com/buy-raspberry-pi

Technical Specification:

Processor

- Broadcom BCM2387 chipset.
- 1.2GHz Quad-Core ARM Cortex-A53 (64Bit)

802.11 b/g/n Wireless LAN and Bluetooth 4.1 (Bluetooth Classic and LE)

- IEEE 802.11 b / g / n Wi-Fi. Protocol: WEP, WPA WPA2, algorithms AES-CCMP (maximum key length of 256 bits), the maximum range of 100 meters.
- IEEE 802.15 Bluetooth, symmetric encryption algorithm Advanced Encryption Standard (AES) with 128-bit key, the maximum range of 50 meters.

GPU

- Dual Core Video Core IV® Multimedia Co-Processor. Provides Open GL ES 2.0, hardware-accelerated Open VG, and 1080p30 H.264 high-profile decode.
- Capable of 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs with texture filtering and DMA infrastructure

Memory

- 1GB LPDDR2

Operating System

- Boots from Micro SD card, running a version of the Linux operating system or Windows 10 IoT

Dimensions

- 85 x 56 x 17mm

Power

- Micro USB socket 5V1, 2.5A

Connectors:

Ethernet

- 10/100 BaseT Ethernet socket

Video Output

- HDMI (rev 1.3 & 1.4)
- Composite RCA (PAL and NTSC)

Audio Output

- Audio Output 3.5mm jack
- HDMI
- USB 4 x USB 2.0 Connector

GPIO Connector

- 40-pin 2.54 mm (100 mil) expansion header: 2x20 strip
- Providing 27 GPIO pins as well as +3.3 V, +5 V and GND supply lines

Camera Connector

- 15-pin MIPI Camera Serial Interface (CSI-2)

Display Connector

- Display Serial Interface (DSI) 15 way flat flex cable connector with two data lanes and a clock lane

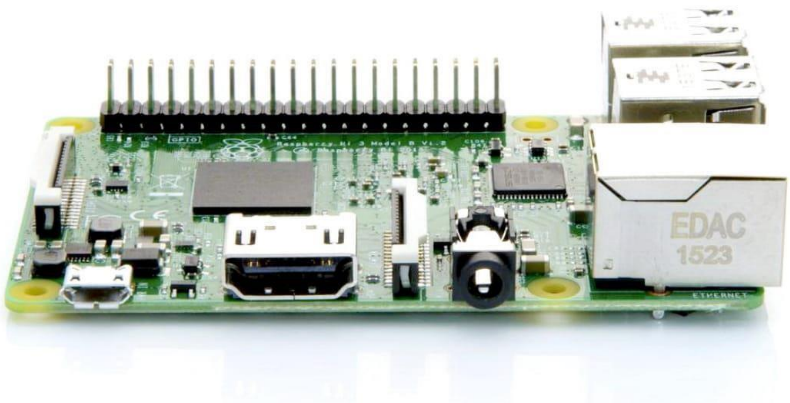
Memory Card Slot

- Push/pull Micro SDIO

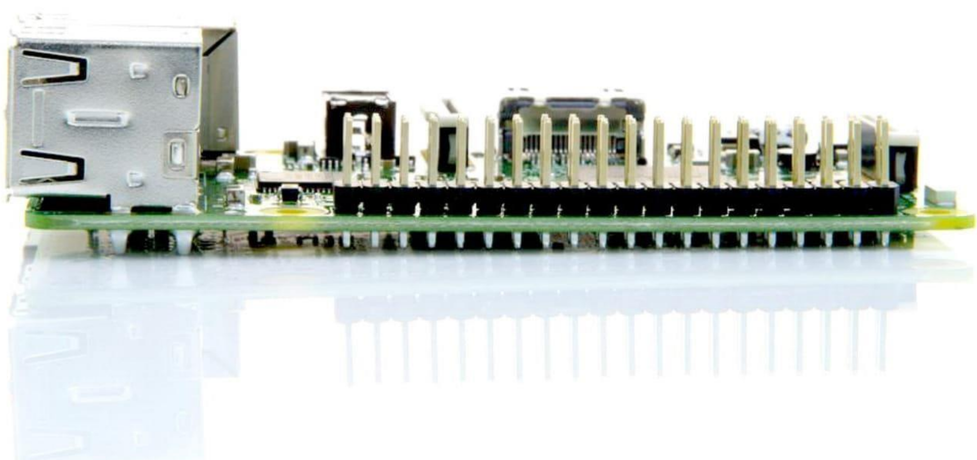
http://uk.farnell.com/buy-raspberry-pi

http://www.newark.com/buy-raspberry-pi

The GPU provides Open GL ES 2.0, hardware-accelerated Open VG, and 1080p30 H.264 high-profile decode and is capable of 1Gpixel/s, 1.5Gtexel/s or 24 GFLOPs of general purpose compute. What's that all mean? It means that if you plug the Raspberry Pi 3 into your HDTV, you could watch BluRay quality video, using H.264 at 40MBits/s



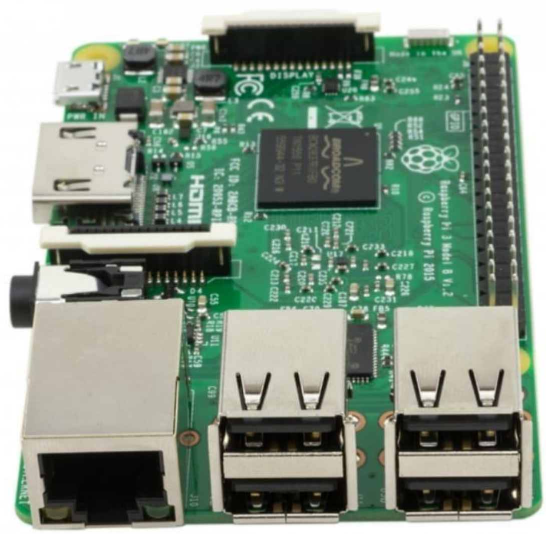
The biggest change that has been enacted with the Raspberry Pi 3 is an upgrade to a next generation main processor and improved connectivity with Bluetooth Low Energy (BLE) and BCM43143 Wi-Fi on board. Additionally, the Raspberry Pi 3 has improved power management, with an upgraded switched power source up to 2.5 Amps, to support more powerful external USB devices.



http://uk.farnell.com/buy-raspberry-pi

http://www.newark.com/buy-raspberry-pi

The Raspberry Pi 3's four built-in USB ports provide enough connectivity for a mouse, keyboard, or anything else that you feel the RPi needs, but if you want to add even more you can still use a USB hub. Keep in mind, it is recommended that you use a powered hub so as not to overtax the on-board voltage regulator. Powering the Raspberry Pi 3 is easy, just plug any USB power supply into the micro-USB port. There's no power button so the Pi will begin to boot as soon as power is applied, to turn it off simply remove power. The four built-in USB ports can even output up to 1.2A enabling you to connect more power hungry USB devices (This does require a 2Amp micro USB Power Supply)



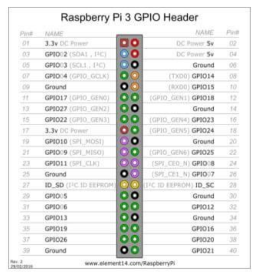
On top of all that, the low-level peripherals on the Pi make it great for hardware hacking. The 0.1" spaced 40-pin GPIO header on the Pi gives you access to 27 GPIO, UART, I²C, SPI as well as 3.3 and 5V sources. Each pin on the GPIO header is identical to its predecessor the Model B+.

SoC

Built specifically for the new Pi 3, the Broadcom BCM2837 system-on-chip (SoC) includes four high-performance ARM Cortex-A53 processing cores running at 1.2GHz with 32kB Level 1 and 512kB Level 2 cache memory, a VideoCore IV graphics processor, and is linked to a 1GB LPDDR2 memory module on the rear of the board.

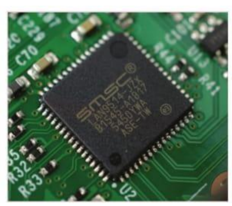
GPIO

The Raspberry Pi 3 features the same 40-pin general-purpose input-output (GPIO) header as all the Pis going back to the Model B+ and Model A+. Any existing GPIO hardware will work without modification; the only change is a switch to which UART is exposed on the GPIO's pins, but that's handled internally by the operating system.



USB chip

The Raspberry Pi 3 shares the same SMSC LAN9514 chip as its predecessor, the Raspberry Pi 2, adding 10/100 Ethernet connectivity and four USB channels to the board. As before, the SMSC chip connects to the SoC via a single USB channel, acting as a USB-to-Ethernet adaptor and USB hub.

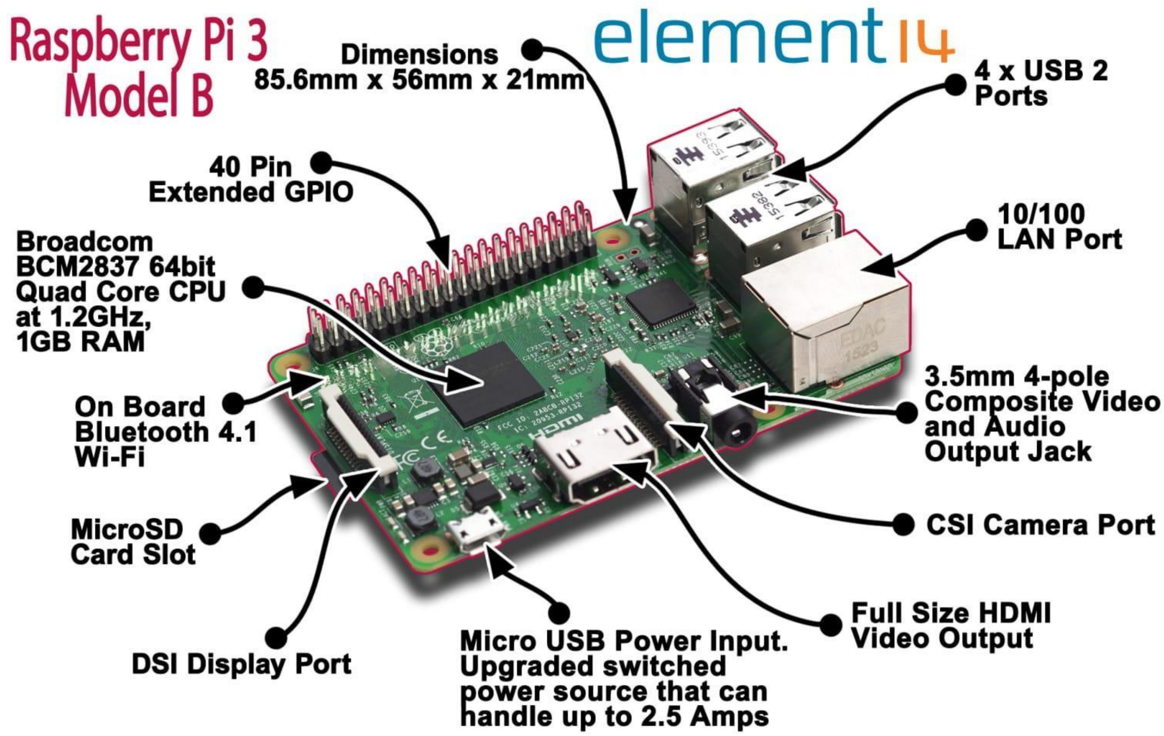


Antenna

There's no need to connect an external antenna to the Raspberry Pi 3. Its radios are connected to this chip antenna soldered directly to the board, in order to keep the size of the device to a minimum. Despite its diminutive stature, this antenna should be more than capable of picking up wireless LAN and Bluetooth signals – even through walls.

http://uk.farnell.com/buy-raspberry-pi

http://www.newark.com/buy-raspberry-pi



Farnell element14 Newark element14

http://uk.farnell.com/buy-raspberry-pi http://www.newark.com/buy-raspberry-pi

Key Improvements from Pi 2 Model B to Pi 3 Model B:

- Next Generation QUAD Core Broadcom BCM2837 64bit ARMv7 processor
- Processor speed has increased from 900MHz on Pi 2 to 1.25Ghz on the RPi 3 Model B
- BCM43143 Wi-Fi on board
- Bluetooth Low Energy (BLE) on board
- Upgraded switched power source up to 2.5 Amps (can now power even more powerful devices over USB ports)

The main differences are the quad core 64-bit CPU and on-board Wi-Fi and Bluetooth. The RAM remains 1GB and there is no change to the USB or Ethernet ports. However, the upgraded power management should mean the Pi 3 can make use of more power hungry USB devices

For Raspberry Pi 3, Broadcom have supported us with a new SoC, BCM2837. This retains the same basic architecture as its predecessors BCM2835 and BCM2836, so all those projects and tutorials which rely on the precise details of the Raspberry Pi hardware will continue to work. The 900MHz 32-bit quad-core ARM Cortex-A7 CPU complex has been replaced by a custom-hardened 1.2GHz 64-bit quad-core ARM Cortex-A53

In terms of size it is identical to the B+ and Pi 2. All the connectors and mounting holes are in the same place so all existing add-ons, HATs and cases should fit just fine although the power and activity LEDs have moved to make room for the WiFi antenna.

The performance of the Pi 3 is roughly 50-60% faster than the Pi 2 which means it is ten times faster than the original Pi.

All of the connectors are in the same place and have the same functionality, and the board can still be run from a 5V micro-USB power adapter. This time round, we're recommending a 2.5A adapter if you want to connect power-hungry USB devices to the Raspberry Pi.

Raspberry Pi 3 Model B

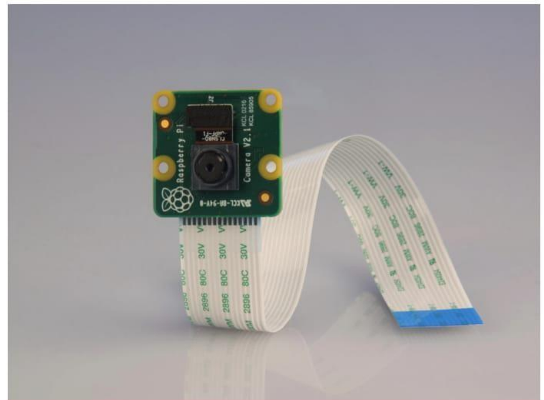
Raspberry Pi 2 Model B

Farnell element14 — http://uk.farnell.com/buy-raspberry-pi

Newark element14 — http://www.newark.com/buy-raspberry-pi

	Raspberry Pi 3 Model B	**Raspberry Pi 2 Model B**	**Model B+**	**Model A+**	**Model A**	**CMDK**
Processor Chipset	Broadcom BCM2837 64Bit ARMv7 Quad Core Processor powered Single Board Computer running at 1250MHz	Broadcom BCM2836 32bit ARMv7 Quad Core Processor powered Single Board Computer running at 900MHz	Broadcom BCM2835 32bit ARMv6 SoC full HD multimedia applications processor	Broadcom BCM2835 32bit ARMv6 SoC full HD multimedia applications processor	Broadcom BCM2835 32bit ARMv6 SoC full HD multimedia applications processor	Broadcom BCM2835 32bit ARMv6 SoC full HD multimedia applications processor
GPU	Videocore IV	Videocore IV	Videocore IV	Videocore IV	Videocore IV	Videocore IV
Processor Speed	QUAD Core @1250 MHz	QUAD Core @900 MHz	Single Core @700 MHz	Single Core @700 MHz	Single Core @700 MHz	Single Core @700 MHz
RAM	1GB SDRAM @ 400 MHz	1GB SDRAM @ 400 MHz	512 MB SDRAM @ 400 MHz	256 MB SDRAM @ 400 MHz	256 MB SDRAM @ 400 MHz	512 MB SDRAM @ 400 MHz
Storage	MicroSD	MicroSD	MicroSD	MicroSD	SDCard	4GB eMMC
USB 2.0	4x USB Ports	4x USB Ports	4x USB Ports	1x USB Port	1x USB Port	1x USB Port
Power Draw / voltage	2.5A @ 5V	1.8A @ 5V	1.8A @ 5V	1.8A @ 5V	1.2A @ 5V	1.8A @ 5V
GPIO	40 pin	40 pin	40 pin	40 pin	26 pin	120 pin
Ethernet Port	Yes	Yes	Yes	No	No	No
Wi-Fi	Built in	No	No	No	No	No
Bluetooth LE	Built in	No	No	No	No	No

Raspberry Pi



Camera Module

Product Name	Raspberry Pi Camera Module
Product Description	High Definition camera module compatible with all Raspberry Pi models. Provides high sensitivity, low crosstalk and low noise image capture in an ultra small and lightweight design. The camera module connects to the Raspberry Pi board via the CSI connector designed specifically for interfacing to cameras. The CSI bus is capable of extremely high data rates, and it exclusively carries pixel data to the processor.
RS Part Numer	913-2664

Specifications

Image Sensor	Sony IMX 219 PQ CMOS image sensor in a fixed-focus module.
Resolution	8-megapixel
Still picture resolution	3280 x 2464
Max image transfer rate	1080p: 30fps (encode and decode) 720p: 60fps
Connection to Raspberry Pi	15-pin ribbon cable, to the dedicated 15-pin MIPI Camera Serial Interface (CSI-2).
Image control functions	Automatic exposure control Automatic white balance Automatic band filter Automatic 50/60 Hz luminance detection Automatic black level calibration
Temp range	Operating: -20° to 60° Stable image: -20° to 60°
Lens size	1/4"
Dimensions	23.86 x 25 x 9mm
Weight	3g

Automated Garbage Segregator

ORIGINALITY REPORT

10%	5%	7%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.slideshare.net Internet Source	1%
2	Mohammed Rafeeq, Ateequrahman, Sanjar Alam, Mikdad. "Automation of plastic, metal and glass waste materials segregation using arduino in scrap industry", 2016 International Conference on Communication and Electronics Systems (ICCES), 2016 Publication	1%
3	"Proceedings of the International Conference on ISMAC in Computational Vision and Bio-Engineering 2018 (ISMAC-CVB)", Springer Nature, 2019 Publication	1%
4	Submitted to Arab American University - Jenin Student Paper	1%
5	Submitted to Tshwane University of Technology Student Paper	1%

www.ijritcc.org

6	Internet Source	<1%
7	Submitted to Universitat Politècnica de València Student Paper	<1%
8	Submitted to University of Greenwich Student Paper	<1%
9	Submitted to Central Queensland University Student Paper	<1%
10	"Hebbian Learning and Negative Feedback Networks", Springer Nature, 2005 Publication	<1%
11	pt.scribd.com Internet Source	<1%
12	www.mcucity.com Internet Source	<1%
13	Loris Nanni, Alessandra Lumini. "Heterogeneous bag-of-features for object/scene recognition", Applied Soft Computing, 2013 Publication	<1%
14	www.groundai.com Internet Source	<1%
15	Afsana Ahsan Jeny, Masum Shah Junayed, Syeda Tanjila Atik, Sazzad Mahamd. "Chapter	<1%

41 A Model for Identifying Historical Landmarks of Bangladesh from Image Content Using a Depth-Wise Convolutional Neural Network", Springer Nature, 2020
Publication

16	Submitted to International Islamic University Malaysia Student Paper	<1%
17	Submitted to Asian Institute of Technology Student Paper	<1%
18	www.ellab.physics.upatras.gr Internet Source	<1%
19	en.wikipedia.org Internet Source	<1%
20	Submitted to Indian Institute of Technology, Madras Student Paper	<1%
21	Yeong-Kang Lai, Chu-Ying Ho, Yu-Hau Huang, Chuan-Wei Huang, Yi-Xian Kuo, Yu-Chieh Chung. "Intelligent Vehicle Collision-Avoidance System with Deep Learning", 2018 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), 2018 Publication	<1%
22	Zakaria Moutakki, Tarik Ayaou, Karim Afdel, Abdellah Amghar. "Prototype of an embedded	<1%

system using Stratix III FPGA for vehicle detection and traffic management", 2014 International Conference on Multimedia Computing and Systems (ICMCS), 2014
Publication

23	"Image Analysis and Processing - ICIAP 2017", Springer Nature, 2017 Publication	<1 %
24	Submitted to Sardar Patel Institute of Technology Student Paper	<1 %
25	Shahram Taheri, Önsen Toygar. "On the use of DAG-CNN architecture for age estimation with multi-stage features fusion", Neurocomputing, 2019 Publication	<1 %
26	Lecture Notes in Computer Science, 2015. Publication	<1 %
27	Nimisha S Gupta, V Deepthi, Maya Kunnath, Pal S Rejeth, T S. Badsha, Binoy C Nikhil. "Automatic Waste Segregation", 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), 2018 Publication	<1 %
28	Sanjay Kumar, Manish Kumar. "A Study on the Image Detection Using Convolution Neural	<1 %

Networks and TenserFlow", 2018 International Conference on Inventive Research in Computing Applications (ICIRCA), 2018
Publication

29	Gallina, P.. "Neural network painting defect classification using Karhunen-Loeve transformation", Optics and Lasers in Engineering, 199907 Publication	<1%
30	info.dcs.warwick.ac.uk Internet Source	<1%
31	www.electronicslovers.com Internet Source	<1%
32	www.city.kitami.lg.jp Internet Source	<1%
33	"Intelligent Information and Database Systems", Springer Nature, 2018 Publication	<1%
34	Guangyao Zhou, Zhiwu Lu, Yuxin Peng. "L1-graph construction using structured sparsity", Neurocomputing, 2013 Publication	<1%

Exclude quotes	On	Exclude matches	< 5 words
Exclude bibliography	On		