International Journal of Engineering Applied Sciences and Technology, 2022
Vol. 7, Issue 3, ISSN No. 2455-2143, Pages 27-33
Published Online July 2022 in IJEAST (http://www.ijeast.com)

FLOWER HARVESTING ROBOT USING COMPUTER VISION

Hawi Teshome, Selamawit chane, Feben Merga
Department of Electrical and Computer Engineering
Addis Ababa science and Technology University, Addis Ababa, Ethiopia

***Abstract*— over the past decades, horticulture industry can be considered as one of the main contributors of the export sector in developing countries like Ethiopia. The horticulture industry deals with plants, mainly for food, beauty and decoration (flowers). This paper aims at providing a system that can improve outdated ways of harvesting methods. Those are mainly manual and laborious tasks which also took a lot of time. In this paper, a harvesting robot is designed and implemented using raspberry pi microcomputer to build a flower harvesting system that can identify, sort and arrange the flowers being harvested without human interaction. Here, a computer vision technology is applied to identify the types and states of the flowers. A convolution neural network image classifier is used along with deep learning networks. The harvesting activity of the machine is assisted by a 3 DOF robotic arm and additional cutting mechanism attached with cutter blade.**

***Keywords*— Horticulture, Raspberry Pi, DOF, Robotic arm.**

I. INTRODUCTION

In flower fields several pesticides are used. Pesticides can cause a short-term adverse health effects, called acute effects, as well as chronic adverse effects that can occur months or years after exposure. This implies that a flower field is a very hazardous place for a human being to be directly exposed to. Also even compared with standard industrial practice, the working environment in greenhouses flower field is very uncomfortable as both the temperature and humidity are too high. So, humans should not be directly involved with this harmful environment. In order to solve these and other problems, in this paper an automatic harvesting system is designed that incorporates various recent technologies like computer vision and deep learning.

Deep learning [1] technologies are becoming the major approaches for natural signal and information processing, such as image classification and speech recognition. Deep learning is a technology inspired by the functioning of human brain. In deep learning, networks of artificial neurons analyze large dataset to automatically discover fundamental patterns, without human intervention, deep learning identify patterns in unstructured data such as, Images, sound, video and text. Convolutional neural networks (CNN) become very popular for image classification in deep learning; CNN‘s perform better than human subjects on many of the image classification datasets [2].

This paper utilizes deep learning on an image classifier CNN for two things: first, to identify the target: whether it is flower or not, and to find out the type of the flower. After the above procedure is completed, the robotic part of the system would carry out the typical human activity of harvesting. That is: cutting and placing the target in the desired spot.

II. RELATED AND EXISTING WORKS

Robotics and advanced production techniques aren‘t just transforming the factory floor; they are also changing the farms. In the 21st century, robotics technology is becoming more and more influential in the way we grow, harvest and process the food we eat.
Harvesting robots appear to be a trending technology in the new era of automation on agriculture. The best example of the newest addition to the high tech greenhouse is a harvesting robot of rosadamascena. This harvesting robot is designed to gather a particular type of flower called ‘rosadamascena’ [3]. As Armin kohan e.t.al suggested the harvesting process is aided by a computer vision technique called stereoscopic machine vision. Since the harvested flowers are desired for industrial purpose the robot doesn‘t emphasize on position of stem. In the paper proposed by Armin Kohan e.t.al the object recognition technique is assisted by two charged couple device (cameras) to capture images. Those images in turn would be used to identify the best image between stereo images.

In the paper suggested [4] by A. Vinoth kumar e.t.al the flower harvesting robot uses image processing technique called water shed. As stated in the paper image processing techniques are achieved as follows ―Image processing is a technique to analyzing and manipulating the image, in charge to get the required in sequence of the picture or to enhance it. In the image processing input is given as a picture and the simulation output might be a picture or some specified characteristics or feature of that image.
A rose harvesting robot is proposed by Prof. Dr. AbdulkadirErden [5]. He anticipated analyzing the rose by a

International Journal of Engineering Applied Sciences and Technology, 2022
Vol. 7, Issue 3, ISSN No. 2455-2143, Pages 27-33
Published Online July 2022 in IJEAST (http://www.ijeast.com)

means of one of computer vision techniques in order to identify whether the rose is harvestable or not.

Finally, when we come to comparing the methods proposed in these papers there have various advantages and short comings. The former need at least three high resolution cameras in order to capture the images from different positions. While the second one requires powerful image processing portable computer in order to manipulate the captured image. The third one proposes to compare the captured images with a large collection of images in the machine database, which in turn requires large memory space on the field machine. After analyzing the methods raised by those scholars this paper proposes a computer vision technique to identify, sort and analyze the captured images. This technique is called "deep learning". Nowadays this method is aspiring technique in various innovations. It is easier to implement and requires lesser resources than the above techniques. The obtained result through this method in this project is also quite satisfactory.

III. METHODOLOGY

Overview

Sectional parts of the system: In order to simplify the prototype, let us classify it in to a number of sectional parts:

A. **Sensory Section-** this section includes an optical sensor or (camera) which continuously fed the system with captured images directly from the environment.

B. **Controlling Center**- this section is like a human brain. It organizes, analyses, and commands other parts of the system with in a small processing time.

C. **Actuation Section-** after analyzing the controller commands to initialize the actuator to take action as per the command.

Fig. 1. Materials used to implement the prototype

Design procedure

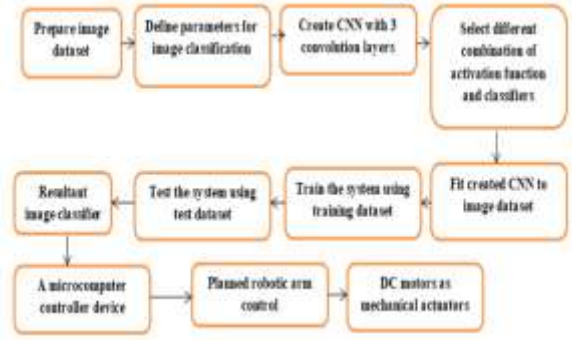


Fig. 2. Image Classifier Models Development Using Software Tools

Before starting to launch the classifier model, some preparation must be done in advance. Those are:

Image Dataset Preparation

A large number of dataset is the key in order to achieve deep learning projects. Datasets ranging up to millions are used in more sophisticated and advanced machine learning application. So, preparing image dataset is the first task in model development. Finding a large number of images for a specific group of interest is not an easy task. Redundancy of images and including irrelevant images is one of the difficulties. Even though there are sites that provides dozens of images at ones, but it choose to prepare a particular datasets for the models.

Fig. 3. Samples of training image sets

The image set must include variety of images from the same class in order to increase the learning rate of the model. Images with various color ranges, frames and back grounds are more effective in achieving high prediction rate. Fig. 3 represents some of sample images from each class. After manually filtering all images one by one, then those images are being classified in to three directories of specific classes. Let us clarify it by using tree structures of the directories prepared for the project.

International Journal of Engineering Applied Sciences and Technology, 2022
Vol. 7, Issue 3, ISSN No. 2455-2143, Pages 27-33
Published Online July 2022 in IJEAST (http://www.ijeast.com)

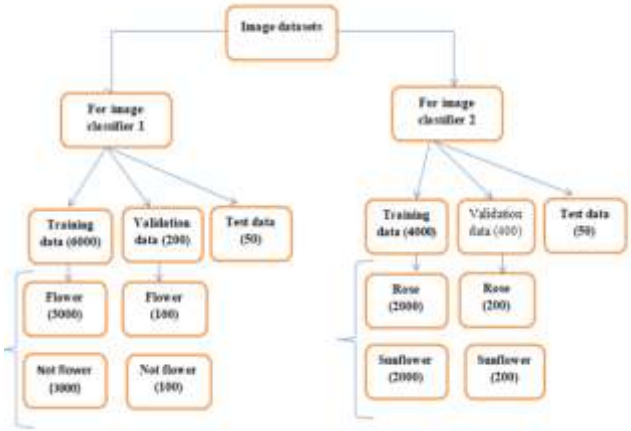


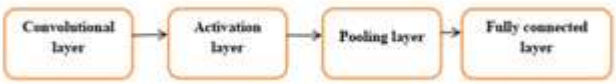
Fig. 4. Image set organization

We use a typical CNN model initialization to launch our image classifiers. Due to comparatively low computational power and low speed of the training device, we are obligated to divide the CNN in to two. The followings are two image classifier models designed for the project.

➢ **Model 1**: flower classifier Categorizes whether the given flower image is 'rose' or 'sunflower' (only two classes for demonstration purpose)

➢ **Model 2**: flower identifier Recognizes whether the given image is flower or not and fed it to model 1.

Both of the models are built using the same approach of building a typical CNN



Steps to Build the Model
1. Importing libraries
2. Loading and preprocessing data
3. Creating a validation set
4. Defining the model structure
5. Training the model
6. Making predictions

The program below is written in python script to initialize the basic sequential model of the CNN.

```
fromkeras.preprocessing.image import
ImageDataGenerator
fromkeras.models import Sequencial
fromkeras.layers import
Conv2D,Maxpooling2D
fromkeras.layers import
Activation,Dropout,Flatten,Dense
```

These lines of the code are useful in order to call powerful deep learning and image processing library packages that are the base stone while building the image classifier model.

```
train_datagen=ImageDataGenerator(
rescale=1./255,
shear_range=0.2,
zoom_range=0.2,
horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1./255)
```

The class Image Data Generator performs the data augmentation. Applying a small amount of the transformations to an input image will change its appearance slightly, but it does not change the class label. So, making data augmentation is a very natural, easy method to apply for computer vision tasks [6]. The Image Data Generator is not returning both the original data and the transformed data the class only returns the randomly transformed data. When the model is being trained we can think of our Image Data Generator class as intercepting the original data, randomly transforming it and then returning it to the neural network for training. For example, the data obtained by data augmented from the original images by applying simple geometric transforms such as random:

- Translations
- Rotations
- Changes in scale
- Shearing
- Horizontal(and in some cases, vertical) flips

```
train_generator=train_datagen.flow_from_directory(train_data_dir,target
_size=(img_width,img_height), batch_size=batch_size,
class_mode='binary')
validation_generator=test_datagen.flow_from_directory(
validation_data_dir,target_size=(img_width,img_height),
batch_size=batch_size, class_mode='binary')
```

The flow_from_directory class accepts the following parameters:

International Journal of Engineering Applied Sciences and Technology, 2022
Vol. 7, Issue 3, ISSN No. 2455-2143, Pages 27-33
Published Online July 2022 in IJEAST (http://www.ijeast.com)

- The directory must be set to the path where your 'N' classes of folders are present.
- The target size is the size of your input images, every image will be resized to this size.
- Batch size: number of images to be yield from the generator per batch
- Class mode: to be set to "binary" since we have only two classes to predict

```
1. model=Sequencial()
2.model.add(Conv2D(32,(3,3),input_shape))
3.model.add(Activation('relu'))
4. model.add(Maxpooling2D(pool_size=(2,2)))
5.model.sumary()
```

In line 1 a sequential model is the basic model of CNN in machine learning package. Then by using the add () method it is possible to add various features. From line 2 to 4 basic features of a CNN are added. Those are: Line 2: convolution layer [7]:

The inputs for the class Conv2D() are:

- Filters: is the number of filters that the convolution will learn from in this case **32** filters will be learnt.
- Kernel_size: it is a list of 2 integers specifying the height and width of the 2D convolution window. In this case a kernel size of 3×3 is chosen
- Data_format: this parameter of Conv2D class be either set to channel first which means the channel (RGB) followed by image width and height or vice versa.

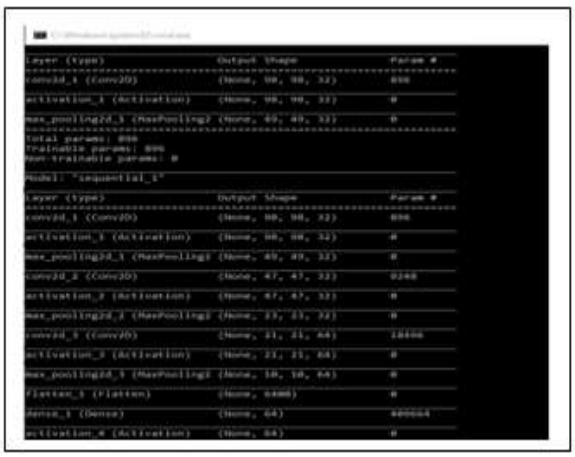


Fig. 5. CNN training progress window

```
1.model.add(Flatten())
2.model.add(Dense(64))
3.model.add(Activation('relu'))
4.model.add(Dropout(0.5))
5.model.add(Dense(1))
6.model.add(Activation('sigmoid'))
```

In line 1 the Flatten() function is used to get a copy of a given array collapsed into one dimension.

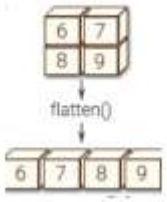


Fig. 6. Horizontal flattening

Dense()

Dense layers learn a weight matrix, where the first dimension of the matrix is the dimension of the input data and the second dimension is the dimension of the output data. In this case; the input layer has a shape 3 for RGB image channel while the output layer has one layer. This means the dense layer will learn a 3×1 weight matrix [8].

Dropout()

The term "Dropout" is used for a technique which drops out some nodes of the network, dropping out can be seen as temporarily deactivating or ignoring neurons of the network. This technique is applied in the training phase to reduce over fitting effects. Over fitting is an error which occurs when a network is too closely fit to a limited set of input samples [9].

Sigmoidal Function

Using mathematical definition, the sigmoid function takes any range real number and returns the output value which falls in the range of 0 to 1. Based on the convention, the output value is expected to be in the range of -1 to 1. The sigmoid function produces and 'S' shape curve [10].

International Journal of Engineering Applied Sciences and Technology, 2022
Vol. 7, Issue 3, ISSN No. 2455-2143, Pages 27-33
Published Online July 2022 in IJEAST (http://www.ijeast.com)

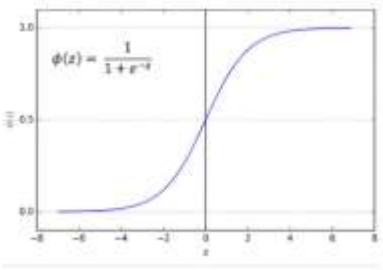


Fig. 7. Sigmoidal Activation Function

```
1.model.compile(loss='binary_crossentropy',optimizer='rmsprop',metrics=['accuracy'])
2.model.fit_generator(train_generator,steps_per_epoch=no_train_samples//batch_size,
epochs=epochs,validation_data=validation_generator,
callbacks=cbks,validation_steps=no_validation_samples//batch size)
```

The training automatically begins when the model. compile method starts

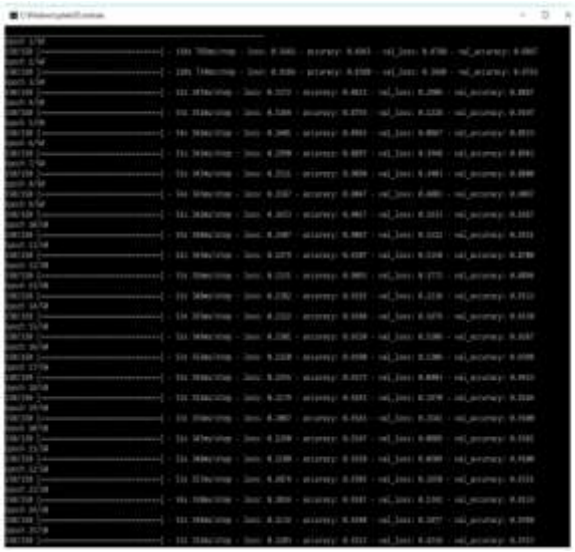


Fig. 8. Training Progress of the Classifier model one

```
model.save('./models/model.h5')
model.save_weights('./models/weights.h5')
```

since the trained model is needed for further use in other machine only the weights which are output of the training are transferred in to the other machine in this case the raspberry pi. So the model and its weights are saved in .h5 format [11].

Hardware Implementation

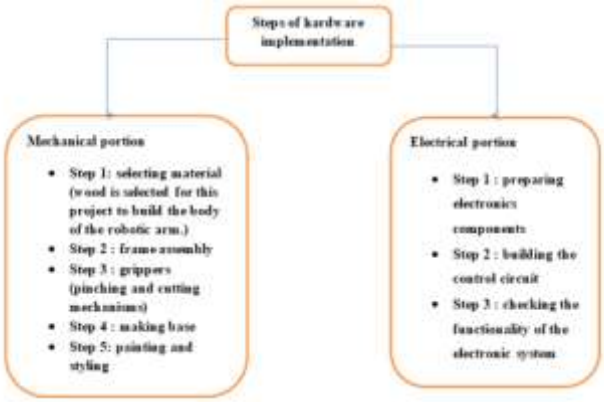


Fig. 9. Hardware Implementation Procedures

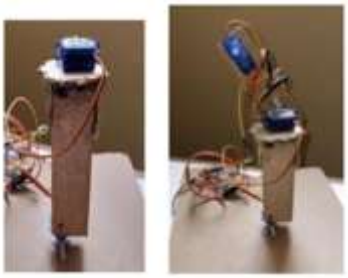


Fig. 10. DIY Robotic arm model

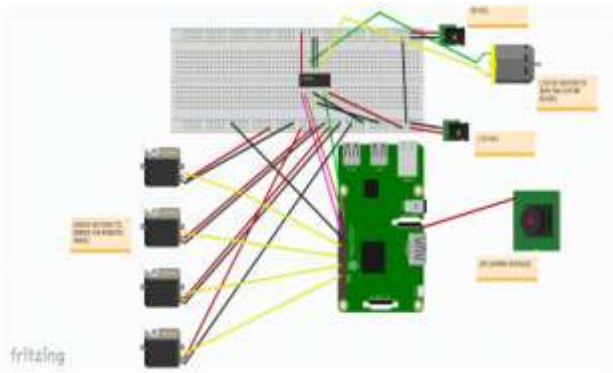


Fig. 11. Electronic connection

As shown on the above image the electrical connection is made discretely as follows:

➢ The ATX power supply is plugged on the wall out let socket. This power supply can provide 5v, 3.3v, 12v, -12v and ground at once. The power needed in this project is 5v for the servos and for motor driver IC whereas +12v to drive the dc motor. According to the color codes of ATX power supply the yellow wire is +12v, black is common and red is 5v. So, those wires will be connected to the bread board using male to male jumper wires [12].

International Journal of Engineering Applied Sciences and Technology, 2022
Vol. 7, Issue 3, ISSN No. 2455-2143, Pages 27-33
Published Online July 2022 in IJEAST (http://www.ijeast.com)

Fig. 12. (a) ATX power supply (b) ATX color code

IV. RESULT AND DISCUSSION

Performance Analysis of the Classifier Models
Model's Accuracy

Here the models' performance is compared with respect to various features. Table-1 indicates that 3 models were trained with different activation function which results in different Accuracy values the two with highest accuracy value are selected for the project. The comparison chart compares these 3 models by epoch size, number of validation and training data sets in Fig. 13.

Table-1 Classifier models accuracy measure

	Model One	Model Two	Model Three
Loss	0.1315	0.1727	0.4434
Accuracy	0.963	0.9400	0.8510
Val. Loss	0.066	0.1133	0.1297
Val. Accuracy	0.9100	0.9264	0.8300
Type of activation function Used for classification	Sigmoidal	Sigmoidal	Softmax

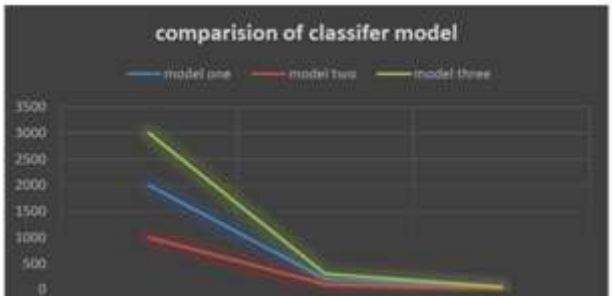


Fig. 13. Comparison Charts between Classifiers

Performance Analysis of the Prototype
Robotic Arm Performance

- A pre-defined constant path is required to be followed for each revolute joint [13].
- Each revolute joint's performance to follow the intended path is evaluated.
- Gripper whether it holds the flower properly or failed to place it on the desired spot.

Table-2 Joints movement

Joints	Starting angle	Approaching angle	Gripping angle	Placing angle
Base	90°	90°	90°	0° for "sunflower" or 180° for "rose"
Elbow	120°	100°	80°	80°
gripper	0°	0°	90°	0°

Fig. 14. (a)Starting position (b) gripping position (c) placing position of robotic arm

Cutter Performance: The cutter can cut any flower stem within radius of 7mm. the cutter is position at approximately 30^o from the horizontal plane. The performance of the cutter elevates in the inclined position than with the 90^o position of the cutter. In the latter case, the flower and the cutter is perpendicular that makes the blade inefficient as a cutter.

Camera Performance: As shown in the figure below even though the image is captured in less lighted environment prediction models succeed in identifying the captured image taken in a much different environment from the training image sets that are taken from a brighter stage.

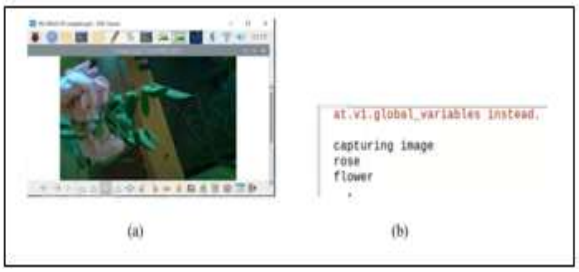


Fig. 15. (a) Captured image from pi camera (b) prediction

V. CONCLUSION

As a final observation, this project, which titled as "flower harvesting robot using computer vision" will improve the gathering of flowers from the older, manual, laborious, and time taking operation to computer vision technology that integrated with harvesting robot. The way of harvesting flowers that is based on robotic and image processing technology can be develop and implement in order to solve the problems associated with traditional way of flower gathering.

Thanks to the robotic arms, many tasks are made not difficult and the resulting error has been reduced to a minimum. And these robotic arms made easier many human intervention and laborious tasks in different area of fields. In this project also, the significance of these robotic arms is great. Despite the fact that the robotic arm made by this project is of prototype quality, it has a caliber that can be improved for more robotic systems. Besides these, robotic arm sector, which is open to development, will keep its importance in the future.
Computer vision will also play a vital role in the development of artificial intelligence by giving them the ability to process information as well as or even better than human visual system. They are applied to identify the types and states of the flowers which are crucial in this project.

VI. REFERENCE

[1] Karan Chaunhan, Shrwarn Ram. (2018). Image Classification with Deep Learning and Comparison between Different Convolution Neural Network Structures using Tensorflow and Keras, International Journal of Advance Engineering and Research Development. vol.5, (pp.1-4).

[2] François Chollet. (2018). Deep Learning with Python, Shelter Island: New York, Manning publication Co. , (pp. 19-24)

[3] Armin Kohan, Ali Mohammad B., MehranYazdi, SaeidMinaei and Mohammad J. . (2011). Robotic Harvesting of Rosa Damascena Using Stereoscopic Machine Vision , WASJ .vol.2, (pp. 37-39)

[4] A.VinothKumar,V.Kannarasu,S.Padmapriya,N.Parthee ban,S.Arun.(2019). Design and Implementation of Autonomous Flower Harvester using Image processing. IJRTE,Volume-8 Issue-2, (pp. 10-15)

[5] CahitGürel, Prof. Dr. AbdulkadirErden. (2013). Conceptual design of a rose harvesting robot for greenhouses. the 20th Int. Conf. on Mechatronics and Machine Vision in Practice. Atılım University, Mechatronics Engineering Dept., (pp. 18-20).

[6] TorbjØrn Grande Østby. (2018). Object Detection and Tracking on a Raspberry Pi using Background subtraction and Convolution Neural Networks, MSc. Thesis, Dept. Natural Sciences and Maritime Sciences. University College of Southeast Norway, (pp. 75-78).

[7] Alvaro Peris. (2020). NMT-Keras Documentation, (pp. 133-134).

[8] J.Petterson and A. Gibson. (2017). Deep learning: A practitioner‘s approach, O‘Reilly Media, (pp. 23-25).

[9] I. Goodfellow, Y. Bengio and A. Courville. (2016) Deep Learning, MIT Press, (pp. 13-17).

[10] Martin Abadi, Paul Barham. (2015). TensorFlow: A system for large-scale machine learning,12thUSENIX Symposium on Operating System Design and Implementation, USENIX Association, (pp.265-270).

[11] Bruno Siciliano and OussamaKhatib. (2008). Springer Handbook of Robotics, Springer Berlin HeidelBerg, (pp.85-87).

[12] Gareth Halfcree. (2010) The official Raspberry Pi Beginner’s guide, Cambridge, Raspberry Pi Trading Ltd, (pp.56-60).

[13] AshwaqAbduameer, Marzipan Sulaiman,M.S.M Aras, DawoodSaleem. (2016). GUI Based Control Cystem Analysis Using PID Controller for Education ,IJEECS. vol.3 No.1, (pp.91-101).