Camera

**Camera Module 2**

Specification

The v2 Camera Module has a Sony IMX219 8-megapixel sensor. The Camera Module can be used to take high-definition video, as well as stills photographs. It’s easy to use for beginners, but has plenty to offer advanced users if you’re looking to expand your knowledge. There are lots of examples online of people using it for time-lapse, slow-motion, and other video cleverness. You can also use the libraries we bundle with the camera to create effects. it’s a leap forward in image quality, colour fidelity, and low-light performance. It supports 1080p30, 720p60 and VGA90 video modes, as well as still capture. It attaches via a 15cm ribbon cable to the CSI port on the Raspberry Pi. The camera works with all models of Raspberry Pi 1, 2, 3 and 4. It can be accessed through the MMAL and V4L APIs, and there are numerous third-party libraries built for it, including the Picamera Python library.

**Raspberry Pi High Quality Camera**

## 12.3 megapixel Sony IMX477 sensor, 7.9mm diagonal image size, and back-illuminated sensor architecture, with adjustable back focus and support for C- and CS-mount lenses.

## Specification

* Sony IMX477R stacked, back-illuminated sensor, 12.3 megapixels, 7.9 mm sensor diagonal, 1.55 μm × 1.55 μm pixel size
* Ouput: RAW12/10/8, COMP8
* Back focus: Adjustable (12.5 mm–22.4 mm)
* Lens standards: C-mount, CS-mount (C-CS adapter included)
* IR cut filter: Integrated
* Ribbon cable length: 200 mm
* Tripod mount: 1/4”-20

Scope

**Hardware Specification**

|  | **Camera Module v1** | **Camera Module v2** | **HQ Camera** |
| --- | --- | --- | --- |
| Net price | $25 | $25 | $50 |
| Size | Around 25 × 24 × 9 mm |  | 38 x 38 x 18.4mm (excluding lens) |
| Weight | 3g | 3g |  |
| Still resolution | 5 Megapixels | 8 Megapixels | 12.3 Megapixels |
| Video modes | 1080p30, 720p60 and 640 × 480p60/90 | 1080p30, 720p60 and 640 × 480p60/90 | 1080p30, 720p60 and 640 × 480p60/90 |
| Linux integration | V4L2 driver available | V4L2 driver available | V4L2 driver available |
| C programming API | OpenMAX IL and others available | OpenMAX IL and others available |  |
| Sensor | OmniVision OV5647 | Sony IMX219 | [Sony IMX477](https://www.sony-semicon.co.jp/products/common/pdf/IMX477-AACK_Flyer.pdf) |
| Sensor resolution | 2592 × 1944 pixels | 3280 × 2464 pixels | 4056 x 3040 pixels |
| Sensor image area | 3.76 × 2.74 mm | 3.68 x 2.76 mm (4.6 mm diagonal) | 6.287mm x 4.712 mm (7.9mm diagonal) |
| Pixel size | 1.4 µm × 1.4 µm | 1.12 µm x 1.12 µm | 1.55 µm x 1.55 µm |
| Optical size | 1/4" | 1/4" |  |
| Full-frame SLR lens equivalent | 35 mm |  |  |
| S/N ratio | 36 dB |  |  |
| Dynamic range | 67 dB @ 8x gain |  |  |
| Sensitivity | 680 mV/lux-sec |  |  |
| Dark current | 16 mV/sec @ 60 C |  |  |
| Well capacity | 4.3 Ke- |  |  |
| Fixed focus | 1 m to infinity |  | N/A |
| Focal length | 3.60 mm +/- 0.01 | 3.04 mm | Depends on lens |
| Horizontal field of view | 53.50 +/- 0.13 degrees | 62.2 degrees | Depends on lens |
| Vertical field of view | 41.41 +/- 0.11 degrees | 48.8 degrees | Depends on lens |
| Focal ratio (F-Stop) | 2.9 | 2.0 | Depends on lens |

**Hardware Features**

| **Available** | **Implemented** |
| --- | --- |
| Chief ray angle correction | Yes |
| Global and rolling shutter | Rolling shutter |
| Automatic exposure control (AEC) | No - done by ISP instead |
| Automatic white balance (AWB) | No - done by ISP instead |
| Automatic black level calibration (ABLC) | No - done by ISP instead |
| Automatic 50/60 Hz luminance detection | No - done by ISP instead |
| Frame rate up to 120 fps | Max 90fps. Limitations on frame size for the higher frame rates (VGA only for above 47fps) |
| AEC/AGC 16-zone size/position/weight control | No - done by ISP instead |
| Mirror and flip | Yes |
| Cropping | No - done by ISP instead (except 1080p mode) |
| Lens correction | No - done by ISP instead |
| Defective pixel cancelling | No - done by ISP instead |
| 10-bit RAW RGB data | Yes - format conversions available via GPU |
| Support for LED and flash strobe mode | LED flash |
| Support for internal and external frame synchronisation for frame exposure mode | No |
| Support for 2 × 2 binning for better SNR in low light conditions | Anything output res below 1296 x 976 will use the 2 x 2 binned mode |
| Support for horizontal and vertical sub-sampling | Yes, via binning and skipping |
| On-chip phase lock loop (PLL) | Yes |
| Standard serial SCCB interface | Yes |
| Digital video port (DVP) parallel output interface | No |
| MIPI interface (two lanes) | Yes |
| 32 bytes of embedded one-time programmable (OTP) memory | No |
| Embedded 1.5V regulator for core power | Yes |

**Software Features**

|  |  |
| --- | --- |
| Picture formats | JPEG (accelerated), JPEG + RAW, GIF, BMP, PNG, YUV420, RGB888 |
| Video formats | raw h.264 (accelerated) |
| Effects | negative, solarise, posterize, whiteboard, blackboard, sketch, denoise, emboss, oilpaint, hatch, gpen, pastel, watercolour, film, blur, saturation |
| Exposure modes | auto, night, nightpreview, backlight, spotlight, sports, snow, beach, verylong, fixedfps, antishake, fireworks |
| Metering modes | average, spot, backlit, matrix |
| Automatic white balance modes | off, auto, sun, cloud, shade, tungsten, fluorescent, incandescent, flash, horizon |
| Triggers | Keypress, UNIX signal, timeout |
| Extra modes | demo, burst/timelapse, circular buffer, video with motion vectors, segmented video, live preview on 3D models |

users can now upload data and train their own custom machine-learning algorithms in the cloud, and then deploy them back to their Raspberry Pi. Four new machine-learning software development kits (SDKs) for Raspberry Pi are available week, including C++, Go, Node.js and Python, allowing users to program their own custom applications for inferencing. Support for object detection has also been added, meaning Raspberry Pi owners can use camera data captured on their device to train their own custom object detection algorithms, instead of having to rely on 'stock' classification models.

* [**Raspberry Pi**](https://www.raspberrypi.org/)**:** A small, affordable computer popular with educators, hardware hobbyists, and robot enthusiasts. Raspberry Pi 3+ or raspberry 4 mother board

TensorFlow is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks. Tensorflow is a symbolic math library based on dataflow and differentiable programming.it is written in python, C++.

The camera for raspberry Pi are cheap, easy to install and connect.

Software

Python

[the best software/applications for your Raspberry Pi?](https://www.raspberrypi.org/forums/viewtopic.php?t=272664#p1652929)

Python  
Mplayer  
Opencv  
Gimp

**How to get the calorie?**

Food Recognition deals with recognition of food item when given an image. For this problem I used Convolutional Neural Network (CNN). A **Convolutional Neural Network (ConvNet/CNN)** is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

1. To detect food type by using Convolutional Neural Network (CNN)
2. To estimate food weight and calories of food.

**How do you code CNN in Python?**

**We have 4 steps for convolution:**

1. Line up the feature and the image.
2. Multiply each image pixel by corresponding feature pixel.
3. Add the values and find the sum.
4. Divide the sum by the total number of pixels in the feature.

Dataset for python

* Use pandas python library.
* **ERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT)**

**[](https://www.ijert.org/)**

Top of Form

Bottom of Form

* **[](https://www.ijert.org/our-indexing)**
* [**HOME**](https://www.ijert.org/)
* [**ABOUT JOURNAL**](https://www.ijert.org/about-journal)
  + [Mission & Scope](https://www.ijert.org/mission-scope)
  + [Editorial Board](https://ems.ijert.org/editorial-board)
  + [Reviewers](https://ems.ijert.org/reviewers)
  + [Peer-Review Policy](https://www.ijert.org/review-process)
  + [Publication Ethics Policy](https://www.ijert.org/publication-ethics-policy)
  + [Journal Policies](https://www.ijert.org/journals-policies)
  + [Indexing](https://www.ijert.org/our-indexing)
  + [Join as Reviewer](https://www.ijert.org/join-as-reviewer)
  + [Conference Partners](https://www.ijert.org/conference-partners)
  + [FAQs](https://www.ijert.org/faqs)
* [**CALL FOR PAPERS**](https://www.ijert.org/call-for-papers)
* [**TOPICS**](https://www.ijert.org/topics)
* [**AUTHORS GUIDE**](https://www.ijert.org/guidelines-for-authors)
  + [Journal Statistics – 2018-2019](https://www.ijert.org/statistics-impact-factor)
  + [Submit Manuscript](https://www.ijert.org/submit-manuscript)
  + [APC – Article Processing Charges](https://www.ijert.org/processing-charges)
  + [Register as Volunteer](https://www.ijert.org/register-as-volunteer)
  + [Downloads](https://www.ijert.org/downloads)
  + [FAQ](https://www.ijert.org/faq)
* [**ARCHIVE**](https://www.ijert.org/published-issue-archive)
* [**CONFERENCES**](https://www.ijert.org/conference-special-issue-proposals)
  + [Upcoming Conferences](https://ems.ijert.org/upcoming-conference-special-issue.php)
* [**CONFERENCE PROCEEDINGS**](https://www.ijert.org/conference-proceedings)
* [**THESIS PUBLICATION**](https://www.ijert.org/publish-dissertation-thesis-online)
  + [Thesis Archive](https://www.ijert.org/thesis-archive)
  + [Thesis Publication FAQs](https://www.ijert.org/thesis-publication-faqs)
  + [Thesis Publication Charges](https://www.ijert.org/thesis-publication-charges)
* [**CONTACT US**](https://www.ijert.org/contact-us)
* [**LOGIN**](https://ems.ijert.org/)
  + [Author Login](https://ems.ijert.org/author-login)
  + [Reviewer Login](https://ems.ijert.org/reviewer-login)

[**IETE - 2020 (VOLUME 8 - ISSUE 11)**](https://www.ijert.org/category/iete-2020-volume-8-issue-11)

**Fruit Recognition System for Calorie Management**

[**DOWNLOAD FULL-TEXT PDF**](https://www.ijert.org/research/fruit-recognition-system-for-calorie-management-IJERTCONV8IS11017.pdf) [**CITE THIS PUBLICATION**](https://www.ijert.org/fruit-recognition-system-for-calorie-management#cite)

* **Open Access**
* Article Download / Views: 471
* **Authors :** Vishnu H S, Sindhushree B, Punith A, Aishwarya K, Praveen G
* **Paper ID :** IJERTCONV8IS11017
* **Volume & Issue :**[IETE – 2020 (Volume 8 – Issue 11)](https://www.ijert.org/iete-2020-volume-8-issue-11)
* **Published (First Online):**04-08-2020
* **ISSN (Online) :** 2278-0181
* **Publisher Name :** IJERT
* **License:**[Creative Commons License](http://creativecommons.org/licenses/by/4.0/) This work is licensed under a [Creative Commons Attribution 4.0 International License](http://creativecommons.org/licenses/by/4.0/)

[PDF Version](https://www.ijert.org/fruit-recognition-system-for-calorie-management)

[**Text Only Version**](https://www.ijert.org/fruit-recognition-system-for-calorie-management)

**Fruit Recognition System for Calorie Management**

Vishnu H S

Electronics and Communication Engineering

RNS Institute of Technology, VTU Bangalore, Karnataka

Sindhushree B

Electronics and Communication Engineering

RNS Institute of Technology, VTU Bangalore, Karnataka

Punith A

Electronics and Communication Engineering

RNS Institute of Technology, VTU Bangalore, Karnataka

Aishwarya K

Electronics and Communication Engineering RNS Institute of Technology, VTU Bangalore, Karnataka

Mr. Praveen G

Assistant Professor

Electronics and Communication Engineering RNS Institute of Technology, VTU Bangalore, Karnataka

AbstractFruits contain important vitamins, minerals and fiber. A diet comprising mainly of fruits and vegetables can help prevent cancer, diabetes and heart diseases. In this paper we execute in an effective type of recognize, fruit recognition is carried out using CNN algorithm. A set of fruit image is trained in a CNN model for recognition a standard nutrition table is referred to measure calorie since using an image to measure calorie is an arduous task. the image is captured by the raspberry pi through the webcam.

KeywordsConvolution Neural Network, Python, Machine Learning, TensorFlow and Keras, image recognition

1. INTRODUCTION

In the recent years, due to a rise in health consciousness, many mobile applications for recording everyday meals have been released. Some of them employ fruit image recognition , which estimates not only fruit names but also fruit calories. However, these applications often require the user to enter information such as fruit categories and size or volume, which render it cumbersome owing to subjective evaluation. To overcome these inconveniences, automatic recognition of the fruit photo is employed on mobile devices. However, in most of the cases, the estimated calories are just associated with the estimated fruit categories or the relative size compared to the standard size of each fruit category which is usually indicated manually by the user. Currently, no applications exist that can estimate fruit calories automatically. Although CNN based image recognition methods have considerably solved most of the image recognition tasks including fruit recognition. fully-automatic fruit calorie estimation from a photo still remains an enigma. This paper proposes a new portfolio of pictures include popular fruits. A portfolio named as fruits. As a second object . a deep intimated neural signal is trained to facilitate identification of fruit image this part can find more larger array of picture It is here that the fruits dataset is described as to how it is created and what its contents are, the aim of the project is developing a application for estimating a fruit calorie and improve peoples consumption conducts for health care.

1. LITERATURE SURVEY

Title : fruit recognition and its calorie measurement : an image processing approach

Using image processing technique the agriculture field and food science field is increasing day by day. The image features are shape, color and texture which is used to classification and calorie estimation of fruits. This paper proposes an algorithm for fruit recognition and its calorie estimation based on the shape, color and texture along with the histogram of gradients and GLCM with the local binary pattern algorithms for texture segmentation scheme recognizing the fruits and area , major axis, minor axis, minor axis is calculated by using the shape feature to get more accurate calorie value. With the help of nutritional look up table these features are fed to multi SVM classifier for accurate classification [15][16]. real time database and pretend plastic fruit databases of MATLAB used for evaluation. Results obtained are very close to real calories of the fruit.

Drawback: the drawback of this paper is it cannot determine raw, ripen and rotten fruit category.

Title: A novel svm based food recognition method for calorie measurement applications

Emerging food classification methods play an important role in food recognition applications. For this purpose, a new recognition algorithm for food is presented, considering its shape, color, size, and texture characteristics. Using various combinations of these features, a better classification will be achieved more importantly. At the classification step the support vector machine method is engaged which leads to better results. Support vector learning is based on simple ideas which originated in statistical learning theory. Supported by our simulation results, the proposed algorithm recognizes food categories with an average approval recognition rate of 92.6%

Drawback:

1. EXPERIMENTAL TOOLS
   1. Hardware requirements

Raspberry Pi

Fig: 2.1 Raspberry Pi-3B

The raspberry pi could be a bargain basement priced, credit-card sized computer which might be easily plugged into a computer monitor or TV. Its capable of doing everything we might expect a desktop IP system to perform, surfing from the online and playing high- definition videos, to creating database data processing, and live games. Raspberry pi has the aptitude to intercommunicate with the outside world, and has been utilized in a spacious array in digital projects, from music equipment and sensors to weather stations and chirping birdhouse with infrared cameras.

Frontech webcam

Fig 2.2 Frontech Webcam

This 20 megapixels webcam comes with an adjustable focus, the frame rate of MOS sensor up to 30fps. To great quality and image. The connectivity of usb 2.o connection allows to simply connect with electronic devices of user, its an mounted microphone. It provides great sound quality.

* 1. Software requirements

Python in machine learning: Python has the library that permits developers to use optimized algorithms. It implements popular machine learning techniques like recommendations, classification, and clustering.

Libraries and packages:

To understand machine learning, you would like to possess basic knowledge of python programming. In addition, there are a number of libraries and packages generally used in performing various machine learning tasks as listed below: numpy: Is used for its N-dimensional array objects. pandas: Is a data analysis library that includes dataframes. matplotlib: Is 2D plotting library for creating graphs and plots.

scikit-learn: The algorithms used for data analysis and data mining tasks.

seaborn: A data visualization library based on matplotlib.

Dataset pre-processing

First to download train and test data sets of fruits. Step 1: setup and install

Step 2: organize your train folder

Step 3: Train your model using our processed dataset Step 4: test your model

1. METHODOLOGY
   1. CNN architecture

Fig: 3.1 CNN architecture

It is encouraged by biological procedures of neurons that resemble the connectivity outline amongst the neurons of the animal visual cortex. The specic cortical neurons reply to incentives only the rare area of the pictorial eld . The pictorial arena of dissimilar neurons moderately overlays such that they cover the pictoria.CNN is a scheme much like a multilayer perception that has been intended for abridged processing requirements. As associated to the other image classication algorithms, it has less pre- processing.

Layers of Convolutional Neural Network

* + 1. Input: This layer holds the raw pixel values of image.
    2. Convolutional Layer: This layer gets te results of the neuron layer that is connected to the input regions. We define the number of filters to be used in this layer. Each filter may be a 5×5 window that slider over the input data and gets the pixel with the maximum intensity as the output.
    3. Rectified Linear Unit [ReLU] Layer: This layer applies an element wise activation function on the image data.
    4. Pooling Layer: This layer perform a down- sampling operation along the spatial dimensions (width, height), resulting in volume.
    5. Fully Connected Layer: This layer is used to compute the score classes i.e. which class has the maximum score corresponding to the input image

fruit

fruit

webcam

webcam

Raspberry pi

Raspberry pi

display

display

* 1. System block diagram

Fig: 3.2 system block diagram

raspberry pi may be raspberry pi defined as single board system its have capability of finding and searching capability it contains video core multimedia GPU, 64bit RISC machine and CPU core The package of the Raspberry Pi Is Raspbian. OpenCV The program is dumped into the Raspberry PI board. Raspbian OS is that the OS involved. The proposed system starts the method by capturing the fruit image. Captured image using raspberry pi through the webcam. Then, the image is transmitted to the processing level in open CV where the fruit features like color shape of fruit samples are extracted. During this project open cv method is employed to detect shape, color of fruit and with the mixture of features the results obtained are very promising. Results shown within the system display. Then come to open CV is an bundle of computer algorithms its contain several sources.

* 1. Design Flow Diagram

Loading dataset

Pre processing

Loading dataset

Pre processing

webcam

webcam

Fruit detection and fruit recognition

Fruit detection and fruit recognition

classification

classification

Calorie estimation

prediction

Calorie estimation

prediction

Fig: 3.3 design flow diagram

Dataset collection: for collecting fruit pictures in data set we elaborate single fruits into different collection. For an different fruit angles and dimension. So, we used to spread subsequent components like camera lighting capture angle, white plate, thumb fruit. Data set have captured image into different angles into different cameras. Large style fruit or large sized dimension fruit fives more comfortable data set and it helps to increase an accuracy of calories which we have done in fruit reorganization of calories measurement. And details of fruit name, number are already stored in dataset. In the process of data set picture divided into many several parts for capturing all angles in all dimensional to get exact perfect calories measurement.

Pre-processing: image cropping note that the pictures from pfid are taken within the laboratory with an enormous white background. it is generally considered good to appear the feature in questions for better ccn results. Thus, we use the function crop from python image library to

process these images. Data augmentation we rotate each time by 45 degree to 315 degree. We gain 18323 images totally. We split the info set into training set, validate set and test set. Hog feature (histogram of gradient) may be a local feature descriptor applied in image processing. The image is split into small regions called cell and intensify gradients are calculated over the within the cells.

Fruit recognition: first, so as to set the accurate answer in our segmentation. An easy exchanges performed on the images captured by us. We can change the image size into desired format to get on accurate efficiency each image compared with size of the picture. If the image is not suitable for a process and its not united with a category. We used a another some technique to compatible with scale of images such are cropping and padding technique. If we apply these technique to these captured image we get

i.e 720×1080 for simplicity. the heavy large image are cropped to the scale before we proceeding to process. Every picture elaborated to various segments of the fruit which been recognized. In this part mainly focused on segmentation to exact images are proposed frequently. And perhaps we used color segmentation , k-mean clustering, texturing tools. And more than our fruit recognition we selected the parts of the process that are cloud svm and deep neural network methodologies. And these are help to get a perfect accuracy of system and last finally we get the calories of the fruit.

Calories Estimation: finally, we want to execute our experimental answers of our part or task of project. Calorie identification of fruits item, as of in the before learning task we explained about our data set into sets. And the answer of 23 task after executing the lesser reduction we are put to type classifier. This is identified output fruit type for a picture or image. Then the expected fruit comes with an extra image. Hence, reduced or lesser were proceed to our learnt size indicator which answer is output approximately size of fruit item.

* 1. Data Flow Diagram

Loading dataset

Train image

Model created

Fit model

Loading dataset

Train image

Model created

Fit model

Fig: 3.4 Level 1

webca m

Input fruit

Fruit detected & fruit recognition

Calorie esti mated

webca m

Input fruit

Fruit detected & fruit recognition

Calorie esti mated

Fig: 3.5 Level 2

Often they're a preliminary step accustomed create an summary of the system which may later be elaborated. Data flow diagrams may be used for the visualization of knowledge processing (structured design). the info flow sheet is additionally called as bubble chart. it's simple graphical formalism that may be accustomed represent a system a system in terms of the input file to the system,

various processing meted out on these data, and therefore the output data is generated by the system.

1. EXPERIMENTAL CONDITIONS

Fig: 4.1 CNN result for fruit detection for Apple

Fig 4.1 following fruit are detected with their name and nutrition value of that object is shown after calculation which is done by Convolutional Neural Network (CNN)

Fig: 4.2 CNN result for fruit detection for Banana

Fig 4.2 following fruit are detected with their name and nutrition value of that object is shown after calculation which is done by Convolutional Neural Network (CNN).

Fig: 4.3 CNN result for fruit detection for Mango

Fig 4.3 following fruit are detected with their name and nutrition value of that object is shown after calculation which is done by Convolutional Neural Network (CNN).

Fig: 4.4 CNN result for fruit detection for Jackfruit

Fig: 4.4 following fruit are detected with their name and nutrition value of that object is shown after calculation which is done by Convolution Neural Network (CNN).

Fig: 4.5 CNN result for unknown object

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Fruit name | Measure | Weight | Calorie |
| 1 | apple | 1 | 182g | 95cal |
| 2 | banana | 1 | 125g | 111cal |
| 3 | mango | 1 | 336g | 202cal |
| 4 | jackfruit | 1(cup) | 151g | 143cal |

Fig: 4.6 Standard Nutrition Table

1. CONCLUSION AND FUTURE WORK

The presentation of program is high and that taken into point of user view from the number of user usage. However the CNNs need an excellent computing system to experiment the large scale of data sets The CNN is able to keep highly nonlinear data. It takes more executable time to coach the signal. And in this performance and speedy matters a lot. Once the program or project well trained, process can be reducing less leads to lower time. The photocopies are properly prepared and every image forms are absolutely checked with CNN. And mostly CNNs are more classified and well suitable for photocopies. When the quality and quantity matter. From our side or mindset

one of the main objects is used for future work for improve an accuracy and perfection in our project. In in future we could add a load cell to calculate the weight of the fruit and also implement freshness detection on the fruit. Another classified objective thing is to elaborate or extend dataset that contain more fruits this will take time to code the process. And we did not taken or included from other paper related to this project.

REFERENCES

1. F. Siddique, S. Sakib, and M. A. B. Siddique, Handwritten Digit Recognition using Convolutional Neural Network in Python with Tensorow and Observe the Variation of Accuracies for Various Hidden Layers, 2019
2. H. Cheng, L. Damerow, Y. Sun, and M. Blanke, Early yield prediction using image analysis of apple fruit and tree canopy features with neural networks, Journal of Imaging, vol. 3, p. 6, 2017
3. I. Sa, Z. Ge, F. Dayoub, B. Upcroft, T. Perez, and C. McCool, Deepfruits: A fruit detection system using deep neural networks,

Sensors, vol. 16, p. 1222, 2016

1. S. Anushadevi, Calorie Measurement of Food from Food Image, International Journal on Applications in Information and Communication Engineering, 2015
2. H. Cheng, L. Damerow, Y. Sun, and M. Blanke, Early yield prediction using image analysis of apple fruit and tree canopy features with neural networks, Journal of Imaging, vol. 3, p. 6, 2017.
3. Y. K. Keiji Yanai, Food image recognition using deep convolutional network with pre-training and ne-tuning, 2015 IEEE International Conference on Multimedia,2105
4. Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, Li Fei-Fei, Large-scale Video Classication with Convolutional Neural Networks. CVPR 2014.
5. Sunjie, Discussion on health monitoring and damage detection of a largespan bridge, IEEE International Conference on Communication, vol 11, 2011.
6. Edward Sazonov, Haodong Li, Darrell Curry, , and Pragasen Pillay, SelfPowered Sensors for Monitoring of Highway Bridges Sensors Journal, IEEE, vol. 9, 2009.
7. Yu-Chieh Chen, A Low-power Design of a Bridge Scour Monitoring System, IEEE International Conference on Communication, vol. 24, no. 1, 2014. K. Elissa,
8. P. Pouladzadeh, S. Shirmohammadi, and A. Yassine, You are what you eat: So measure what you eat!, IEEE Instrumentation & Measurement Magazine, vol. 19, no. 1, pp. 915, 2016.
9. Qiang Fu, Bridge Vibration Monitoring System Based on Vibrating- Wire Sensor and ZigBee Technologies, 9th IEEE International Conference on Communication Software and Networks ,2017.
10. G. Cowburn and L. Stockley, Consumer understanding and use of nutrition labelling: a systematic review, Public Health Nutrition, vol. 8, no. 01, 2005.
11. Andrew Gastineau, Tyler Johnson,Arturo Schultz Bridge Health Monitoring and Inspections A Survey of Methods vol. 11, Issue 7, September 2009.
12. Parameshachari B D et. al Secure Transfer of Images Using Pixel Level and Bit Level Permutation Based on Knight Tour Path Scan Pattern and Henon Map1st International Conference on Recent Trends in Electronics & Communication Engineering (ICRTECE)organized by REVA university in association with TIE on 11-12 June 2020.
13. Prabu, S., M. Lakshmanan, and V. Noor Mohammed. "A multimodal authentication for biometric recognition system using intelligent hybrid fusion techniques." Journal of medical systems 43.8 (2019): 249.

# Raspberry Pi High Quality Camera

1. 12.3 megapixel Sony IMX477 sensor, 7.9mm diagonal image size, and back-illuminated sensor architecture, with adjustable back focus and support for C- and CS-mount lenses

## **Specifications**

* Sony IMX477R stacked, back-illuminated sensor, 12.3 megapixels, 7.9 mm sensor diagonal, 1.55 μm × 1.55 μm pixel size
* Ouput: RAW12/10/8, COMP8
* Back focus: Adjustable (12.5 mm–22.4 mm)
* Lens standards: C-mount, CS-mount (C-CS adapter included)
* IR cut filter: Integrated
* Ribbon cable length: 200 mm
* Tripod mount: 1/4”-20

## Software For the Raspberry Pi High Quality Camera

* To use any of the official cameras with a Raspberry Pi we first need to enable the camera interface in the Raspberry Pi Configuration tool (sudo raspi-config)l. Then, after a reboot, we can start taking pictures.

You may be thinking that we could use the Raspberry PI High Quality Camera with applications such as Cheese or guvcview, but sadly this is not easily done, and in our tests the images created with Cheese had a nasty purple tint, and guvcview would refuse to start with the camera connected. Hopefully, future software updates will resolve these issues. The first time we take a picture with any official camera, we use raspistill, Raspbian’s built-in capture command, which can quickly show if the camera is connected and working as expected. To test the focus of the camera we ran raspistill with the -k switch, which enabled us to set the focus of the lens.

The Raspberry Pi High Quality Camera’s maximum resolution is 4056 x 3040 pixels (5K) and this produces an image of around 6MB in size. Images are typically saved as JPG, but we can also select RAW, GIF, BMP, PNG, YUV420, RG8888 file formats.

Another way to capture images and video is via the Picamera Python module which enables the camera to be used in projects powered by the popular programming language. With Picamera we can trigger the camera to take an image / video using sensors and inputs connected to the GPIO. At the time of writing the Picamera library has basic functionality and can record 1080p video and 5K images, and also manipulate the images on the fly. Enhanced functionality for the Raspberry Pi High Quality Camera is still a little way off yet, developers are working to make this happen.

**Image Quality of the Raspberry Pi High Quality Camera**

Advertisement

The previous camera modules provided a decent entry into photography with Raspberry Pi, but they images that were at best “good”. With a fixed focus lens, the subject had to be brought to the lens and composition was cramped. Version 2 of the Raspberry Pi Camera Module did feature a focus ring to alter the focus of the camera. But this required a special tool to rotate, otherwise you'd risk scratching the lens. Color reproduction was poor and image quality was grainy. But for less than $30 (sometimes less than $10) what did we expect?

With the Raspberry Pi High Quality Camera, we see much sharper images with rich colors. A manually controlled focus means that we can compose our shots and focus in on specific areas and blur out the rest of the shot for that professional look.

We took the same shots using the older V2 camera and the new High Quality Camera and as you can see the images are vastly superior with the new camera.

Overall the image quality with the Raspberry Pi High Quality Camera is much better. Indoors and out, we were able to take clear images and change the light levels manually and focus our shots according to the subject. Sadly the weather was not with us at the time of review; rain and electronics rarely mix so further testing will be conducted and added to the review.

To record video on the Raspberry Pi High Quality Camera,  we can use the Python Picamera library or we can use **<CODE>raspivid</CODE>** in the terminal. The video output from the latter is in H264 format, which can be played using the VLC or omxplayer media players. At the time of writing the playback is not perfect, and can jump around erratically from time to time. But playing the same video on our Lenovo laptop provided a smooth experience.

So can the Raspberry Pi High Quality Camera capture 4K video? Sorry, no. The best is still 1080p at 30 fps. We did push the camera a little further and managed to get 1080p at 60fps, but it was a little glitchy. To capture higher fps videos we need to drop the resolution. We captured decent footage at 800x600 120fps but it was a little glitchy. We found that converting the video to m4v via the excellent Handbrake video conversion tool provided much better playback.