

## TRATAMIENTO DE IMÁGENES UTILIZANDO OPENCV Y RASPBERRY PI 4

Mtro. Luis Alberto Flores Montaña  
Email: lfloresm1703@alumno.ipn.mx  
Mtro. Rodrigo Vázquez López  
Email: rodrigo.em2@hotmail.com

Instituto Politécnico Nacional  
Centro de Innovación y Desarrollo Tecnológico en  
Cómputo

Dr. Jacobo Sandoval Gutiérrez  
Email: j.sandoval@correo.ler.uam.mx

Universidad Autónoma Metropolitana Unidad  
Lerma  
Departamento de Procesos Productivos

y

Boletín No. 84  
1o. de mayo de 2021

### Resumen

La tecnología encargada del tratamiento de imágenes es primordial en la actualidad para diversas operaciones dentro de distintos sectores de la población, como es el caso de la industria y entretenimiento. El uso de esta última por lo general utiliza cámaras sofisticadas y sistemas computacionales costosos; sin embargo, se pueden utilizar dispositivos económicos y de bajo consumo computacional. Dichas características se pueden encontrar en las microcomputadoras Raspberry pi 4, las cuales son adecuadas para funcionar con diversos tipos de hardware y software y con distintos lenguajes de programación.

En el contenido de este escrito se considera la instalación de software especializado en el tratamiento de imágenes llamado OpenCV, el cual es instalado en una tarjeta Raspberry Pi 4; adicionalmente se hace uso de una cámara conectada a este dispositivo, conocida como PiCamera. Con estas herramientas se realiza la identificación de diversos objetos donde se detecta su forma y color, así como la identificación de diversos rostros haciendo uso de algoritmos especializados.

**Palabras Clave:** Raspberry pi 4, OpenCV, PiCamera, Reconocimiento, Contornos.

Abstract

The technology in charge of image processing is essential today for various operations within different sectors of the population, such as industry and entertainment. The use of the latter usually uses sophisticated cameras and expensive computer systems; however, inexpensive, low-power devices can be used. These features can be found in Raspberry pi 4 microcomputers, which are suitable to work with various hardware and software and with different programming languages.

In the content of this writing, the installation of specialized software in image treatment called OpenCV, which is installed on a Raspberry Pi 4 card, is considered; additionally, use is made of a camera connected to this device, known as PiCamera. With these tools, various objects are identified where their shape and color are detected, as well as the identification of various faces using specialized algorithms.

**Keywords:** Raspberry pi 4, OpenCV, PiCamera, Recognition, Contours.

## I. Introducción

Actualmente el uso de la tecnología para identificar patrones es diverso, se utilizan distintos algoritmos y herramientas para llevar a cabo con éxito la identificación de estos.

En este proyecto se utiliza una variedad de herramientas para la identificación de diversos patrones, tal es el caso de una microcomputadora Raspberry Pi 4, una cámara adaptable a dicho dispositivo llamada PiCamera, así como el uso del lenguaje de programación Python y una biblioteca de visión artificial conocida como OpenCV. Este último incluye también algoritmos adecuados para la identificación de distintos objetos.

Dentro de la literatura se tienen diversas investigaciones referentes al tratamiento de imágenes haciendo uso de OpenCV. En el año 2014 un estudio se centró en el diseño e implementación de una cámara de seguridad inteligente de bajo costo con capacidad de visión nocturna usando Raspberry Pi y OpenCV. El sistema fue diseñado para ser utilizado dentro de un almacén, teniendo la capacidad de detección humana y de humo que puede proporcionar precaución ante posibles delitos e incendios; adicionalmente se envía alarmas de sonidos grabados como: "intruso." "humo detectado", posteriormente a la detección se envían imágenes capturadas al correo electrónico del usuario a través de Wi-Fi [1].

En el año 2017 se propuso un sistema de automatización del hogar inteligente, especialmente para personas mayores, basado en Python, OpenCV, Raspberry Pi y aplicaciones de Android. Los dispositivos están controlados por el servidor Raspberry pi, que funciona según el comando del usuario (táctil o de voz) recibido desde el teléfono móvil. Un sistema de monitoreo de puerta único está diseñado en función de la detección y el reconocimiento de rostros desde una cámara instalada fuera de la puerta principal, a la que se puede acceder desde el teléfono mediante la aplicación de Android [2].

En el año 2019, se utilizó una Raspberry Pi y módulos de cámara relacionados para la adquisición de imágenes. Donde se presta especial atención a la grabación de imágenes estereoscópicas y software de posprocesamiento que aplica OpenCV. Con un diseño de una red de cámaras y se desarrolla a una aplicación de campo. La biblioteca de visión por computadora de OpenCV y su enlace de Python proporciona algunas muestras de secuencias de comandos para alentar a los usuarios a desarrollar sus propias secuencias de comandos personalizadas [3].

## II. Materiales y métodos

### Modelado

Como ya se había mencionado, el proyecto está basado en la realización de identificación de distintos patrones, con el uso de una tarjeta Raspberry Pi 4, y con una cámara, mejor conocida en el medio como PiCamera (estos dispositivos se muestran en la figura 1), se desarrollan distintos experimentos, así como procedimientos para la instalación de la PiCamera y OpenCV.



---

**Figura 1.** Raspberry Pi 4 y PiCamera.

---

Cabe mencionar que una cámara en la Raspberry Pi es imprescindible para cualquiera que cuenta con una tarjeta Raspberry pi 4, u otro modelo, que como ya se ha mencionado se pueden generar diversos proyectos interesantes.

La Raspberry Pi 4 ofrece un puerto dedicado para añadir una extensión de cámara. Al estar integrado en la placa principal como se muestra en la figura 2, donde la PiCamera, ya está instalada, cabe mencionar que se requiere que la cinta lectora debe estar en una posición en especial ya que de otra manera no es posible que la tarjeta pueda leer las placas.



---

**Figura 2.** Puerto de la cámara en la Raspberry pi 4.

---

Cabe mencionar que para la instalación en cuanto al software se hizo bajo un sistema operativo oficial llamada Raspberry Pi OS, el cual dicho sistema es el más reciente para las tarjetas versión 4.

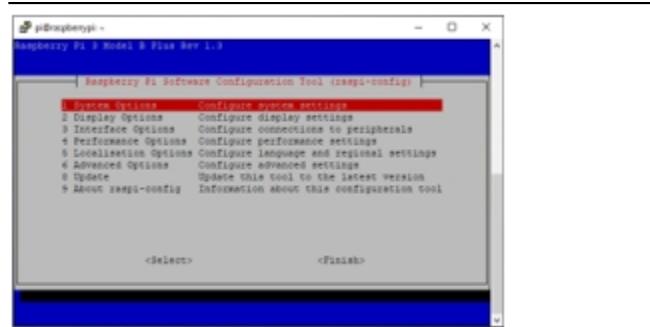
Antes de utilizar la cámara en la tarjeta Raspberry Pi, se necesita habilitar en el sistema operativo; ya que por lo general el puerto de la cámara está desactivado. Por lo tanto, se debe de iniciar la herramienta raspi-config (se puede iniciar desde una terminal), por lo que se escribe el siguiente comando:

---

```
sudo raspi-config
```

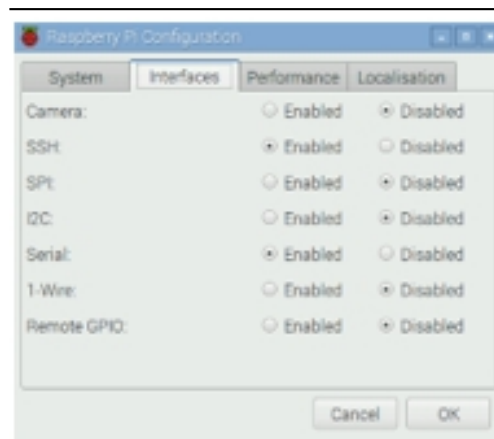
---

A continuación, se despliega la siguiente pantalla mostrada en la figura 3.



**Figura 3.** Herramienta de configuración de software.

Posteriormente se da en "Opciones de interfaz" > "Cámara", y seleccionar habilitar (enable en inglés). Como se muestra en la figura 4.



**Figura 4.** Configuración de interfaces.

Finalmente se debe de salir de "raspi-config" y dar en aceptar en el recuadro reinicio (sale como ventana emergente).

Instalación de OpenCV en el sistema

Para la instalación de OpenCV en la Raspberry Pi, se puede seguir un método estándar de instalación y probarlo.

Los beneficios de utilizar OpenCV en la versión 4.1.0 son las optimizaciones de tiempo de ejecución habilitadas para un conjunto más grande de funciones en módulos core e imgproc.

El módulo DNN (en inglés deep neural network) tiene varias mejoras:

1. El back-end de Inference Engine se ha cambiado a la API NN Builder, compatible con Intel® Neural Compute Stick 2.
2. Reducción del consumo máximo de memoria y compatibilidad con varias redes nuevas de TensorFlow.

- Se agregaron métodos de calibración mano-ojo.
- Se agregaron varios mapas de colores secuenciales perceptualmente uniformes.

Para el proceso de instalación, se debe de asegurar de que la Raspberry Pi esté colocada en un ambiente fresco para evitar el sobrecalentamiento, lo que podría ralentizar el proceso.

A continuación, se muestra los pasos a seguir para la instalación de OpenCV, cabe mencionar que ya se debe de tomar en cuenta la instalación de Raspberry Pi OS y la habilitación e instalación de la PiCamera en la tarjeta.

**Paso 1:** Actualización de Raspberry Pi OS a la última versión.

Abrir una terminal y actualizar la lista de paquetes de raspberry pi:

```
sudo apt-get update && sudo apt-get upgrade && sudo rpi-update
```

**Paso 2:** Aumentar el tamaño de intercambio (en inglés swap-size). El intercambio es un archivo en el disco que sirve como espacio RAM "desbordado". De forma predeterminada, el tamaño de intercambio de Raspberry Pi es de 100 Mb, que es demasiado pequeño, ya que al instalar OpenCV, su proceso se bloqueará innecesariamente. Para evitar esto, necesitamos aumentar el tamaño del intercambio. Para aumentar el tamaño de intercambio, se abre el archivo de intercambio de la siguiente manera:

```
sudo nano /etc/dphys-swapfile
```

y edita la variable

```
CONF_SWAPSIZE :  
#CONF_SWAPSIZE=100  
CONF_SWAPSIZE=2048
```

**Paso 3:** Instale herramientas y bibliotecas para openCV

```
sudo apt-get install build-essential cmake pkg-config  
sudo apt-get install libjpeg-dev libtiff5-dev libjasper-dev  
libpng12-dev  
sudo apt-get install libavcodec-dev libavformat-dev libswscale-  
dev libv4l-dev  
sudo apt-get install libxvidcore-dev libx264-dev  
sudo apt-get install libgtk2.0-dev libgtk-3-dev  
sudo apt-get install libatlas-base-dev gfortran
```

**Paso 4:** Instalar el Python3 y pip3 (En dado caso que no esté instalado el Python3) Si no tiene Python instalado, puede instalarlo con el siguiente comando:

```
sudo apt-get install python3-dev  
sudo apt-get install python3-pip
```

### Paso 5: Obtener el código fuente de OpenCV 4.1.0

Se descarga y descomprime de la siguiente manera:

```
wget -O opencv.zip  
https://github.com/opencv/opencv/archive/4.1.0.zipwget -O  
opencv_contrib.zip  
https://github.com/opencv/opencv\_contrib/archive/4.1.0.zipunzip  
opencv.zipunzip opencv_contrib.zip
```

Puede verificar la última versión en el sitio oficial y actualizar los comandos y las rutas anteriores.

### Paso 6: Instalar numpy

Numpy se usa para realizar operaciones de matriz en Python.

```
sudo pip3 install numpy
```

### Paso 7: Compilar OpenCV

Para este paso, se debe crear una carpeta de compilación donde se crean todos los archivos.

```
cd ~/opencv-4.1.0/  
mkdir build  
cd buildcmake -D CMAKE_BUILD_TYPE=RELEASE \  
-D CMAKE_INSTALL_PREFIX=/usr/local \  
-D INSTALL_PYTHON_EXAMPLES=ON \  
-D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib-4.1.0/modules \  
-D BUILD_EXAMPLES=ON ..
```

Si está intentando instalar una versión diferente de OpenCV, actualice las rutas en consecuencia.

### Paso 8: Compilar OpenCV

Este es el paso más importante y puede llevar incluso más de 3 horas. Para usar los cuatro núcleos en la Raspberry Pi, se debe escribir el siguiente comando:

```
make -j4
```

Asegúrese de que la raspberry pi complete este paso sin errores.

### Paso 9: Instalar OpenCV

Ahora, finalmente se puede instalar OpenCV.

```
sudo make install && sudo ldconfig
```

En este momento, si todo se hace sin errores, se puede reiniciar el sistema

```
sudo reboot
```

**Paso 10:** abre la consola de Python y se importa la biblioteca como se muestra en la figura 5.>

```
~/raspistill: ~$ python3
Python 3.5.3 (default, Sep 27 2018, 17:25:39)
[GCC 6.3.0 20170516] on linux
Type "help()" for more information.
>>> import cv2
>>> cv2.__version__
4.1.0
>>>
```

**Figura 5.** Importación de biblioteca OpenCV a Python .

### III Resultados

#### Tomar una foto

Lo primero que se hizo para probar la conexión de la cámara fue tomar una s foto de la imagen vista por la cámara, esto se hace a través del comando "raspistill".

Dicho comando ya está instalado en el sistema operativo.

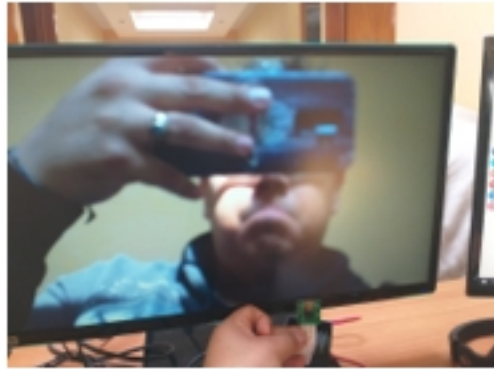
Para usarlo, la línea de comando básica es:

```
raspistill -o image.jpg
```

Con -o se define el nombre del archivo de destino (donde se guardarán las imágenes). Por otra parte, es posible utilizar un nombre de archivo que incluya la ruta, por ejemplo:

```
raspistill -o ~/Pictures/mypicture.jpg
```

En la figura 6, se muestra un ejemplo de como fue posible tomar una captura de imagen en la pantalla haciendo uso de la PiCamera



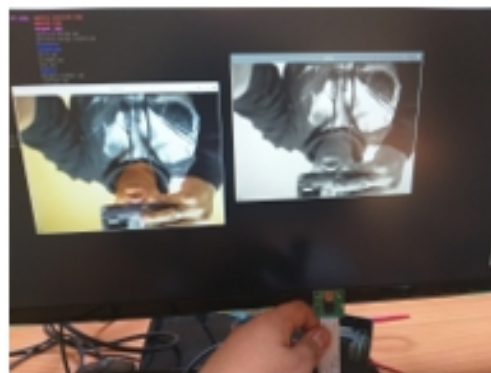
**Figura 6.** Ejemplo de captura de imagen con la cámara PiCamera.

Grabar un vídeo

Para poder grabar un vídeo, el comando a utilizar es "raspivid". El cuál es el siguiente:

```
raspivid -o video.h264
```

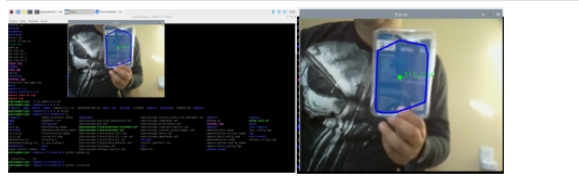
Por otro lado, se realizó un programa donde se muestra la transformada de escalas de grises en tiempo real en la figura 7 se muestra el contraste de las imágenes:



**Figura 7.** Transformada de escala de grises.

Identificación de objetos con color

Para la identificación de objetos es posible con diversas funciones de OpenCV en conjunto con Python, como el caso de `cvtColor` y `drawContours` y `convexHull`, con éstas funciones es posible detectar objetos de acorde a su color, dibujar los contornos y adicionalmente dimensionar el tamaño, como se muestra en la figura 8 donde se tomó un empaque de color azul, donde se dibuja el contorno con líneas de color verde y donde también se muestra las dimensiones.



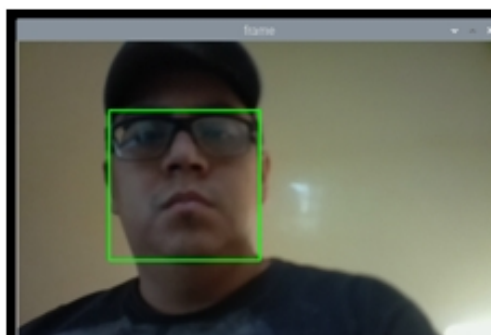
**Figura 8.** Identificación del color de un objeto .

#### Detección de rostros con el algoritmo Haar-Cascade y OpenCV

Una prueba experimental con OpenCV viene con un entrenador y un detector. Sin embargo; se puede entrenar un propio clasificador para cualquier objeto como es el caso de un automóvil, aviones, etc., que utilizan OpenCV.

Aquí la detección con OpenCV contiene muchos clasificadores previamente entrenados para rostro, ojos, sonrisa, etc. Los archivos XML se almacenan en la carpeta `opencv / data / haarcascades /`. Por lo que se crea un detector de rostro y ojos con OpenCV.

Primero se necesita cargar los clasificadores XML requeridos. Luego cargar la imagen de entrada (o video) en modo de escala de grises. Después se encuentran las caras en la imagen. Si se encuentran caras, devuelve las posiciones de las caras detectadas como Rect (x, y, w, h). Cabe destacar que es importante tener el archivo XML (que contiene en el algoritmo de nombre "haarcascade\_frontalface\_default.xml"), en el mismo lugar donde se tiene el archivo a compilar de lo contrario arrojará un error al momento de ejecutarlo. El resultado se muestra en la figura 10, donde hace la detección del rostro.



**Figura 9.** Detección de rostros con el algoritmo Haarcascade .

#### VI. Conclusiones

Este documento muestra de manera sencilla la instalación de OpenCV y Python; así como los procedimientos de identificar diferentes objetivos en video a través de una PiCamera. También es importante mencionar que la tarjeta Raspberry pi 4 tiene limitantes en el poder de procesamiento por eso es importante aumentar el tamaño de intercambio (en inglés `swape_size`); sin embargo, es importante volver a cambiar el valor que tiene por defecto, esto con la finalidad de evitar un sobrecalentamiento en la tarjeta.

Por otro lado, cabe destacar que ya existen algoritmos previamente entrenados e incluidos dentro las carpetas de instalación de OpenCV, como es el caso de los algoritmos pertenecientes a la familia de haarcascade; sin embargo, se puede entrenar distintos algoritmos propios basándose en estos o en alguna otra familia de algoritmos dedicados a la identificación de distintos patrones.

Referencias y recursos electrónicos

1. International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management, Institute of Electrical and Electronics Engineers., IEEE Computational Intelligence Society., Neuronemech (Firm),, Institute of Electrical and Electronics Engineers., International Symposium on Computational Intelligence and Intelligent Informatics, & Engineering Research and Development for Technology Consortium. **(2014)**. *Título del artículo* 2014 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM): November 12-16, 2014, Hotel Centro, Puerto Prinsesa, Palawan, Philippines.
  
2. ] International Conference on Intelligent Computing and Control Systems, IEEE Electron Devices Society,, Institute of Electrical and Electronics Engineers., & Vaigai College of Engineering,. **(2018)**. *Título del artículo* Proceeding of the 2018 International Conference on Intelligent Computing and Control Systems (ICICCS) June 14-15, 2018.
  
3. ] International Society for Photogrammetry and Remote Sensing. **(2011)**. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Science*
  
4. Ashwin Pajankar. **(2015)**. *Raspberry Pi Computer Vision Programming* Packt Publishing.