PRESENTACIÓN

Diap 3: Comenzaremos instalando el ambiente Raspbian para poder comenzar a trabajar con nuestra Raspberry pi.
Comenzamos descargando en nuestra computadora el Raspberry pi Imager para poder grabar nuestra tarjeta uSD con Raspbian
Posteriormente, cargamos la versión que mejor nos acomode, en este caso y para evitar conflictos mayores que se puedan presentar, utilizaremos la versión "Buster" de 32bits, pero es importante mencionar que hasta hoy día, la versión más actualizada es "Bullseye".
Al terminar la escritura en nuestra uSD, podremos conectarla a la entrada de nuestra Raspberry, así como conectar la Raspberry a un monitor, mediante el cable HDMI y conectarlo a una fuente mediante el cable de fuente de poder.
Diap4: Para continuar con la configuración, como la implementación de nuestra red wifi, conectamos el teclado y mouse inalámbricos.

Diap7: Comenzamos abriendo la terminal, antes de instalar OpenCV en su Raspberry Pi 4:
La fundación Raspberry Pi ha lanzado recientemente un software nuevo y mejorado para estas EEPROM. Esto no tiene nada que ver con OpenCV, sino más bien con la disipación de calor. En una de nuestras aplicaciones de visión, el calor de la CPU cae de 65 °C (149 °F) a 48 °C (118 °F) simplemente actualizando el contenido de las EEPROM. Y, como sabe, una temperatura baja de la CPU prolongará la vida útil de su Pi. Para obtener más información, consulte este artículo .
Verificaremos y si es necesario, actualice las EEPROM con los siguientes comandos.
Para el desarrollo de este proyecto utilizaremos OpenCV lanzó la versión 4.5.5. pues es la versión que nos permite eliminar el mayor número de errores a la hora de programar y compilar en Python, pero hay un problema a mencionar. El script de instalación de Python3 se modifica incorrectamente. Colocará todas las bibliotecas en el directorio incorrecto y Python no las encontrará. Al agregar -D PYTHON3_PACKAGES_PATH=/usr/lib/python3/dist-packages a la compilación, las bibliotecas se almacenan en la carpeta correcta.

El chip de RAM físico es utilizado tanto por la CPU como por la GPU. En una Raspberry Pi 2 o 3, el valor predeterminado es de 64 Mbytes asignados para la GPU. El Raspberry Pi 4 tiene un tamaño de memoria GPU de 76 Mbytes. Puede resultar algo pequeño para proyectos de visión, por lo que lo cambiaremos a 128 Mbyte. Después de esta acción, el sistema quiere reiniciarse.
Diasp8: El siguiente paso es aumentar su espacio de intercambio. OpenCV necesita mucha memoria para compilar. Las últimas versiones quieren ver un mínimo de 6,5 GB de memoria antes de construir. Su espacio de intercambio está limitado a 2048 MB por defecto. Para superar este límite de 2048 MByte, deberá aumentar este máximo en el archivo
Ahora vamos a instalar opencv, tecleando en la terminal script de instalación que ejecuta todos los comandos de esta guía a la vez. Toda la instalación tardará una hora y media en completarse
Al terminar, simplemente verificamos que la instalación haya sido exitosa tecleando: Python3 >>Import cv2 >>cv2.__version__ Si todo está correcto, debe aparecer de esta forma

Diap9: Imutils es una serie de funciones de conveniencia para acelerar la computación OpenCV en la Raspberry Pi. Tecleamos el siguiente comando en la terminal de la Raspberry pi: sudo pip3

install imutils

Tensorflow es una plataforma de aprendizaje automático de código abierto. Facilita la creación de modelos de aprendizaje automático para computadoras de escritorio, dispositivos móviles, la web y la nube, sin importar si eres principiante o experto. Para la instalación de TensorFlow, tecleamos el siguiente comando en la terminal: sudo pip3 install https://github.com/lhelontra/tensorflow-onarm/releases/download/v2.1.0/tensorflow-2.1.0-cp37-none-linux_armv7l.whl Yo ya lo tenía instalado por lo que sólo verificamos que esté dentro de Python.

Diap10: Instalar Adafruit_CircuitPython_CharLCD y Adafruit_Blinka. Es fácil usar una pantalla LCD de caracteres con CircuitPython o Python y el módulo Adafruit CircuitPython CharLCD . Este módulo le permite escribir fácilmente código Python que controla una pantalla LCD de caracteres (ya sea retroiluminación simple o retroiluminación RGB). Puede usarlos con cualquier placa de microcontrolador CircuitPython o con una computadora que tenga GPIO y Python gracias a Adafruit_Blinka, nuestra biblioteca de compatibilidad CircuitPython-for-Python. En este código hacemos uso de los paquetes de Python3 Adafruit_CircuitPython_CharLCD y Adafruit_Blinka. Para instalarlos ejecuta los siguientes comandos en la terminal:

En este caso, ya se encuentran instalados, por lo que aparece del modo anterior.

Detección de cámara Después de instalar las librerías pertinentes, se hará la detección de la cámara externa, iniciamos habilitando la cámara en nuestra Raspberry.
Seguido de ello, detectamos el puerto donde se encuentra la cámara, mediante el código de lsusb

Diap11: Al tener detectada la cámara mostraremos más información del puerto USB con el comando: lsusb -d0c45:64ab -v

Se agregará el comando sudo apt-get install luvcview para que instale el visor de la cámara USB en la Raspberry y ejecutamos la cámara con el comando luvcview

DIap12: Al hacer zoom infinito en una imagen se podría observar que está conformada de pequeños recuadros a los cuales llamamos pixeles, que es la construcción fundamental de cualquier imagen, cuando decimos que una imagen tiene resolución de 1280x1200 quiere decir que tiene 1280 pixeles en el eje de las “x”, y 1200 en el eje de las “y”. Un píxel es un valor numérico, en el caso de una imagen en blanco y negro, el pixel es un valor entre 0 y 255, (blanco 0 y negro 255) y lo de en medio es la escala de grises. Cuando las imágenes son a color, los pixeles se forman de tres superpuestos, uno en color roja, verde y azul. Y al hacer juegos entre cada uno de ellos genera toda la gama de colores.

Para que un algoritmo de IA aprenda una clasificación, necesita las características, de esta matriz se extraen esas características mediante la convolución, esta consiste en super poner una imagen sobre la imagen que tenemos, de forma que solo dejara pasar alguna serie de pixeles. Es un filtro.

Con ello podremos obtener: El tamaño de los filtros es siempre menor al de la imagen, y se aplican de manera progresiva, es decir si tenemos una imagen de 1000 pixeles, y el filtro es de 3x3 pues el filtro se aplicará de 3 x3 hasta haber recorrido toda la imagen.

Diap13: Creamos una base de datos (dataset) con dos clases, por un lado, tendremos imágenes a color de rostros con mascarillas, y en otro directorio imágenes de rostros de personas sin mascarillas. La cual fue obtenida de Github

Diap 14: CODIGO

Diap17: Al ejecutar nuestro entrenamiento, se generará un archivo XML que será nuestro "ejecutable" para el programa principal.
diap21: Abrimos un script de Python, y tecleamos el siguiente código:

Diap23: utilizaremos el paquete de python Adafruit_CircuitPython_CharLCD para controlar la pantalla de manera rápida y con pocas líneas de código.
En este código hacemos uso de los paquetes de Python3 Adafruit_CircuitPython_CharLCD y Adafruit_Blinka. Para instalarlos ejecuta los siguientes comandos en la terminal:

Este módulo le permite escribir fácilmente código Python que controla una pantalla LCD de caracteres (ya sea retroiluminación simple o retroiluminación RGB).

Puede usarlos con cualquier placa de microcontrolador CircuitPython o con una computadora que tenga GPIO y Python gracias a Adafruit_Blinka, nuestra biblioteca de compatibilidad CircuitPython-for-Python .

Para poder visualizar el texto sin mascarilla en nuestra pantalla LCD, proseguimos con: Circuito

Diap24: Funciones de los GPIO →Pueden ser configurados tanto como entrada como de salida. →Los pines GPIO también pueden ser activados y desactivados mediante código. Es decir, pueden ponerse a 1 (alto nivel de voltaje) o a 0 (bajo nivel de voltaje). →Por supuesto pueden leer datos binarios, como los unos y ceros, es decir, señal de voltaje o ausencia de ella. →Valores de salida de lectura y escritura. →Los valores de entrada pueden ser configurados en algunos casos como eventos para que generen algún tipo de acción sobre la placa o sistema. Algunos sistemas embebidos los usan como IRQ. Otro caso es configurar que cuando uno o varios pines estén activos por ciertos sensores realice alguna acción… →En cuanto al voltaje e intensidad, debes saber bien las capacidades máximas aceptables por la placa, en este caso la Raspberry Pi 4 o la 3. No debes pasarlos para no dañarla.

HACKS:

Un **buffer** es un espacio de memoria en el que se almacenan datos evitando que el programa que los necesita se quede sin datos durante una transferencia. Los datos son almacenados en un **buffer** mientras se transfieren desde un dispositivo de entrada o antes de enviarlos a un dispositivo de salida.