EVALUACIÓN FUNCIONAL DE LA AIY VISION KIT DE GOOGLE

PRESENTADO POR:

LUIS JOSE CARO DIAZ

UNIVERSIDAD DEL SINÚ ELÍAS BECHARÁ ZAINÚM SECCIONAL CARTAGENA

FACULTAD DE CIENCIAS EXACTAS E INGENIERÍA

ESCUELA DE INGENIERÍA DE SISTEMAS

CARTAGENA – COLOMBIA

EVALUACIÓN FUNCIONAL DE LA AIY VISION KIT DE GOOGLE

Trabajo de grado presentado como requisito para optar el título de

INGENIERO DE SISTEMAS

Asesor disciplinar

LUIS FERNANDO MURILLO FERNÁNDEZ

Asesor metodológico

EUGENIA ARRIETA RODRÍGUEZ

UNIVERSIDAD DEL SINÚ ELÍAS BECHARÁ ZAINÚM SECCIONAL CARTAGENA

FACULTAD DE CIENCIAS EXACTAS E INGENIERÍA

ESCUELA DE INGENIERÍA DE SISTEMAS

CARTAGENA – COLOMBIA

ACTA DE CALIFICACIÓN Y APROBACIÓN

Nota de aceptación:

_____

_____

_____

_____

_____

Director de Escuela

_____

Director de Investigaciones

_____

Firma del jurado

_____

Firma del jurado

Cartagena de Indias.

TABLA DE CONTENIDO

TABLA DE ILUSTRACIONES

AGRADECIMIENTOS.

Primeramente, le doy gracias a Dios y a todas esas personas que me brindaron su ánimo y apoyo, ya, que, sin ello no habría sido posible terminar este trabajo. También le quiero dar las gracias y reconocimiento al tutor de este proyecto, Luis Fernando Murillo Fernández, por su apoyo, dedicación, paciencia y por toda su gran ayuda, Gracias por todos sus consejos a la hora de desarrollar este proyecto, como por su esfuerzo en corregir y perfeccionar el trabajo. En segundo lugar, dar las gracias a todos esos compañeros con los que he convivido durante estos años en la carrera de ingeniería de sistemas y que me enseñaron cada uno algo que me hicieron mejorar tanto estudiantil, profesionalmente y sobre todo como persona. Y por último y no menos importante, dar las gracias a mis papas, mis hermanos, mi esposa y en especial a mi hijo recién nacido, ya que gracias a todas estas personas que siempre estuvieron ahí y que sin ellos nada de esto sería posible.

RESUMEN

En este trabajo se podrá analizar las diferentes características y distintas funcionalidades que tiene la cámara de AIY VISION KIT de Google, y los beneficios que esta tiene en el ámbito académico, específicamente en el proceso de aprendizaje de los estudiantes de la escuela de ingeniería de sistemas de la universidad de Sinú en futuros desarrollos y aplicar de una manera sencilla y didáctica la inteligencia artificial, o en su defecto a lo que se refiere a la visión artificial para proyectos futuros. Las características que más se manejaran en este proyecto son; la captura de rostro del ser humano y la opción de reconocimiento de objetos, estas dos opciones ayudaran al estudiante a clasificar cualquier rostro o cualquier objeto que esté en su entorno.

Es de resaltar que en este mismo proyecto además de analizar y probar todos estos algoritmos, también se podrá estudiar los componentes de hardware y de software, esto para que las personas que quiera seguir con esta investigación tengan una base sólida y concreta desde donde tiene que comenzar su nueva investigación.

INTRODUCCIÓN

En la actualidad se observa un gran avance de la industria cinematográfica respecto a la producción de mejores animaciones y efectos de video. También se escucha hablar de robótica colaborativa e industria 4.0, dichas tecnologías están ligadas a la visión por computadora, la cual permite obtener información de los objetos que existen en el entorno, saber dónde se encuentran, si están en movimiento o no y que trayecto siguen; se puede determinar un camino libre de obstáculos, es decir, se logra una descripción de los objetos físicos que son captados por la cámara. Por esta razón la visión por computadora es el eje principal al que se apunta con el proyecto de evaluación funcional de la AIY VISIÓN KIT de Google que puede ser utilizado para desarrollar prototipos de captura de movimiento del cuerpo humano basado en video con elementos de bajo costo. Es importante resaltar que lo relacionado con la captura de movimiento del cuerpo humano mediante cámara, apunta al termino Mocap (Motion capture o captura del movimiento) que es usado para referirse a aquellas técnicas de grabación de movimiento del cuerpo humano, que captura datos espaciotemporales y representados digitalmente.

Cuando se avanza con la investigación referente a métodos y técnicas de captura de movimiento del cuerpo humano mediante video, se encuentran muchos métodos y técnicas que contribuyen al desarrollo de un algoritmo que permita la aprehensión de movimiento mediante cámara web y herramientas de bajo costo. La investigación respecto a las diferentes técnicas y algoritmos que existen para la captura de movimiento mediante cámara permiten adentrase en el mundo de la cinematografía, televisión, videojuegos y robótica, puesto de muchas de estas industrias ya utilizan procedimientos estandarizados, que colaboran con el desarrollo de este proyecto aportando nuevos conocimientos en el área de la investigación y desarrollo de tecnologías de visón artificial que actualmente tienen un costo elevado y que en Colombia son muy poco conocidas.

1. DISEÑO METODOLÓGICO

En este primer fragmento de trabajo, se verán los puntos más interesantes y puntuales que nos ayudara a desarrollar satisfactoriamente y a buen término nuestra tesis, aquí veremos como, por ejemplo. la descripción del problema, los objetivos, marco referencial y la metodología con la que se va a trabar

1.1. Descripción del Problema

La robótica colaborativa se puede definir como aquellas maquinas capaces de interactuar con los seres humanos y que no exista peligro alguno, en la actualidad estas colaboraciones o interacciones se pueden ver inmersas en compañías y empresas como la NASA y grandes cadenas hoteleras a nivel mundial, la robótica colaborativa hace parte de industrias 4.0(cuarta revolución industrial), ya que involucra inteligencia artificial entre otras tecnologías, puede verse su uso en los procesos de producción para la optimización y agilizar los procesos de los mercados.

En la robótica colaborativa los robots son caracterizados por ser ligeros, flexibles y fáciles de instalar, y han sido diseñados especialmente para interactuar con humanos en un espacio de trabajo compartido sin necesidad de instalar vallas de seguridad, pero muchas personas aún creen que trabajar con robot es algo peligroso y que se corre muchos riesgos, ya que al chocar o tropezar con una de estas máquinas se puede generar grandes lesiones, mutilaciones e incluso hasta la muerte, una solución a este problema es lograr que la maquina logre identificar la posición de cuerpo en el entorno de trabajo, con la intención que la maquina se anticipe a los movimientos del ser humano y así evitar colisiones futuras cuando ambos cuerpos (humano y robot) se encuentren cerca, es en esta parte donde la captura de movimiento y la visión artificial juegan un papel fundamental.

La cuarta revolución industrial, conocida como industria 4.0 es un nuevo paradigma donde convergen una serie de tecnologías como visión artificial, Inteligencia

artificial, Big Data, reconstrucción 3D, realidad aumentada y otras. Colombia en este momento se encuentra en una transición desde la tercera hacia la cuarta revolución industrial, lo cual presenta grandes oportunidades, pero a la vez grandes desafíos para los ingenieros.

Dentro de la iniciativa de industria 4.0 se menciona un conjunto disperso de tecnologías necesarias para conseguir sus planteamientos. Todas ellas convergen en el Internet of Things (IoT), como eje fundamental, pero también se presentan una serie de tecnologías clave necesarias como el Big Data, la automatización industrial, la gestión del ciclo de vida de producto, los dispositivos inteligentes, la ciberseguridad y las tecnologías semánticas. Las tecnologías que se mencionan, en la mayoría de los casos, usan el Visual Computing como medio integrador" [1].

La visión por computadora se ha venido posicionando como herramienta de apoyo para sistemas de vigilancia inteligente, sistemas de reconocimiento y conteo de objetos, identificación de personas, análisis de calidad en sistemas de fabricación, sistemas de apoyo diagnóstico. Se considera en la actualidad que es una de las ramas tecnológicas que tendrá un gran impacto en el desarrollo futuro, y es clave que los ingenieros de sistemas posean habilidades en el desarrollo, configuración y puesta en servicio de estos sistemas.

El programa de ingeniería de sistemas de la Universidad del Sinú - Cartagena no tiene en la actualidad sistemas de visión por computadora basados en SBC (Single Board Computer) para el desarrollo de sistemas de robótica colaborativa que permitan a los estudiantes, docentes e investigadores adquirir habilidades y conocimientos en estas tecnologías de cuarta revolución industrial.

Una tecnología clave en la presente revolución industrial (la denominada Industria 4.0) es la de los robots colaborativos, en los cuales personas y robots comparten su espacio de trabajo, haciendo posible la combinación de la fuerza, velocidad y repetitividad de los robots industriales con la adaptabilidad del trabajador humano,

abriendo con ello nuevas posibilidades y mayor flexibilidad a la producción industrial [2].

En la actualidad la apropiación de las nuevas tecnologías por parte del ser humano en diferentes áreas de trabajo es muy extensa, desde el inicio formal de la automatización y la robótica, hace casi 50 años, se ha planteado a los robots trabajando junto con los humanos, pero desde finales de los años 90s se ha trabajado el concepto de robótica colaborativa, que hace referencia a la idea de humanos y robots compartiendo el mismo espacio de trabajo. Cuando se da la popularización del uso de los robots industriales, se pensaba que los robots remplazarían por completo a los humanos en muy corto plazo, la realidad es que existen muchas tareas en los procesos que aun requieren la interacción humana directa. Desde hace algunos años la robótica colaborativa se ha introducido a las industrias, apuntando a la necesidad de las empresas, fundamentalmente en las áreas de fabricación y logística, de ganar competitividad en el mercado y buscando optimizar los procesos de producción.

Una solución a la interacción entre las máquinas y las personas es lograr que las máquinas identifiquen la posición en el espacio de trabajo, de las personas; con el fin de poder anticiparse a los movimientos de la persona, y de esta manera el dispositivo tome acciones que eviten una colisión o impacto, cuando exista la aproximación de una parte del cuerpo de la persona a la máquina.

Es aquí, donde los sistemas de captura de movimiento ejercen un papel importante, y entre estos, los sistemas de visión artificial son de los más utilizados. Uno de los más grandes inconvenientes en la masificación del uso de estas tecnologías, es el costo de los sistemas, debido al uso de sensores, cámaras, y en general, hardware de muy altas especificaciones y alto precio.

1.2. Justificación

La visión por computadora (CV) como un campo de la Inteligencia Artificial (IA), permite la obtención, el procesamiento y análisis de información de cualquier tipo, captada a través de imágenes digitales, es así como nacen los sistemas de captura de movimiento, capaces de detectar diversas características, tales como la distancia, posición con respecto al espacio, velocidad y dirección del movimiento y más, de una persona u objeto. En las aplicaciones de industria 4.0 y especialmente en la robótica colaborativa, la visión artificial permite implementar sistemas de seguridad, telemando y HMI (*Human Machine Interface*).

Los sistemas actualmente en el mercado en general están desarrollados a partir de una cámara que integra el software y el hardware necesario para la captura y procesamiento de imágenes, pero la mayoría de estos sistemas suele costar miles de dólares, lo cual significa una inversión inmensa si se quiere instalar a cada máquina.

Hace 20 años surgió una biblioteca o librería de visión por computadora conocida como *OpenCv* (*Open Source Computer Visión*) desarrollada por Intel, caracterizada por ser de código abierto y que en el transcurso de los años ha tenido una evolución sorprendente para múltiples aplicaciones como en sistemas de seguridad con detección de rostro, sistemas de seguimiento de objetos y otros sistemas que requieren de visión artificial.

Hoy en día, el desarrollo de sensores de bajo costo y de buenas prestaciones, así como la popularización de sistemas de desarrollo de SoC (*System on Chip*) empotrados de costo razonable, genera una tendencia a desarrollar sistemas de visión por computadora utilizando este hardware de bajo costo, y aprovechando la disponibilidad de herramientas software de código abierto.

En esta fase inicial del proyecto se obtendrá un prototipo del sistema a partir de un kit desarrollado por AIY junto con Google y que a utilizando un SBC (*Single Board Computer*) *Raspberry* Pi, es capaz de desarrollar sistemas de visión por computador

Se espera que en futuras fases del proyecto se pueda implementar este sistema en la interacción de una persona con un robot, utilizando técnicas de robótica colaborativa.

Adicionalmente esta herramienta permitirá a los estudiantes del programa de ingeniería de sistemas de la Universidad del Sinú - Cartagena, adquirir competencias y habilidades en el desarrollo de sistemas de visión por computadora.

1.3. Formulación del problema

¿Cómo desarrollar sistemas de visión por computadora que permitan la integración sencilla con sistemas de robótica colaborativa?

1.4. Alcance

Se pretende realizar una evaluación funcional mediante pruebas a un prototipo de sistema de visión por computadora a partir de un kit AIY de Google, las pruebas y el entrenamiento de dicho algoritmo se realizarán en un entorno de luz controlada (laboratorio de electrónica de la universidad del Sinú).

1.5. Objetivos

1.5.1. General

Evaluar funcionalmente un prototipo de sistema de visión por computadora para apoyo en sistemas de robótica colaborativa mediante el kit AIY Google Visión.

1.5.2. Específicos

- Diseñar una aplicación del sistema de sistema de visión por computadora para apoyo en sistemas de robótica colaborativa utilizando el kit AIY Google Visión.

- Desarrollar una aplicación del sistema de sistema de visión por computadora para apoyo en sistemas de robótica colaborativa utilizando el kit AIY Google Visión.
- Verificar el sistema de sistema de visión por computadora para apoyo en sistemas de robótica colaborativa mediante pruebas para la validación de los resultados.

1.6. Estado del arte

Los sistemas de captura de movimiento basados en video tienen un campo amplio en la industria del cine, robótica y video juegos, ya que en la actualidad se ha incrementado el interés y el uso de estos sistemas, debido a la disponibilidad de métodos de procesamiento de imágenes de buen prestigio, así mismo como el uso de sensores y cámaras de bajo costo. También se ha aumentado el número de proyectos enfocados a la visión por computadora y el uso práctico de estos métodos en la vida cotidiana para el desarrollo e implementación de esta tecnología en robótica colaborativa, por lo que se mencionan los siguientes proyectos y estudios enfocados a la visión por computadora o visión artificial.

El estudio denominado: “Navegación de robot móvil usando Kinect, OpenCV y Arduino”[1], cuyo objetivo fue el desarrollo de un sistema de apoyo a la navegación de un robot móvil por medio de imágenes de profundidad y el reconocimiento de objetos por sus canales primarios. Permitió ver como se utiliza el sensor Kinect de Microsoft para realizar la captura de la imagen RGB y la imagen de profundidad, este sensor consta de una cámara RGB y un emisor de infrarrojos que proyecta un patrón irregular de haces con una intensidad variable. Se logra ver como el sensor de profundidad reconstruye una imagen a partir de la distorsión del patrón, buscando todos los puntos rojos de la escena, contándolos, calculando el centroide, el diámetro, la posición y la distancia a la que se encuentra el objeto que se desea captar respecto a la ubicación del Kinect. Por último, se analiza como procesan los datos y como se realiza la toma de decisión de movimiento para ser enviada al

Arduino, donde se controlan los motores. se estudian los resultados obtenidos en la investigación, los cuales indican que las imágenes de profundidad capturadas por el Kinect requieren de escenarios con iluminación controlada.

En el estudio denominado “Captura de movimiento utilizando el Kinect para el control de una plataforma robótica controlada de forma remota por medio de seguimiento de los puntos de articulación del cuerpo”[2]. La apertura de mercados y la visualización de una calidad de vida más amena y accesible a todos, hace que cada día crezcan las líneas de tecnología con miras a convertirse en aliadas del ser humano para toda actividad y más si esta coloca en riesgo la preservación de la especie humana como tal. Se analiza como en áreas donde se liberen gases, espacios confinados, terrenos minados y desastres naturales; se hace necesaria la interacción natural entre el hombre y la máquina; convirtiéndose el robot en una aplicación moderna para interactuar en diversas circunstancias con las personas. También se observa como a lo largo del tiempo se han desarrollado métodos de captura de movimiento, los cuales han sido empleados en áreas como la medicina, la educación, la seguridad entre otros y vemos como de esta manera se presenta en el mercado el sensor Kinect, dispositivo diseñado con tal propósito inicialmente, lo cual se fue diversificando a lo largo que fueron presentándose códigos abiertos para acceder y focalizar más aplicaciones con él. Vemos como en este proyecto se hace la integración del Kinect con una línea bastante robusta como es la robótica, utilizando el software ROS, haciendo uso de controladores y paquetes para obtener la esqueletización y visualización de Joints. El paquete *Skeleton_Marker* que tiene como característica reconocer un usuario y realizar la esqueletización generando un solo ID, también se hizo uso del controlador *Nite* para obtener la imagen de profundidad que garantiza la estabilidad de los joints. Se observa que una vez obtienen la imagen, pasan a la etapa de creación de un nuevo paquete para recibir la información generada por los puntos censados; para esta tarea implementan *Transform Frame* de ROS (TF), que hace relación a una transformación de los datos generados para ser enviados; en esta secuencia se utilizó los mensajes de ROS

(listener) para enviar y recibir la información de un paquete a otro. y para que lograran una visión del comportamiento del *Joints* se implementan por código la toma de 100 muestras de cada punto, vistas por terminal en los tres ejes coordenados; y 10 muestras reales las cuales se hicieron con un medidor laser, con el fin de relacionar las medidas generadas por el Kinect y un instrumento de medida. Por último, se analiza como con la obtención de todos los datos, realizan el tratamiento estadístico para presentar el grado de error a nivel de joints y global tomando el esqueleto.

En la tesis “Desarrollo de una aplicación de detección de movimiento basada en comparativa estructural de imágenes” [3], se observa el desarrollo de una aplicación de detección de movimiento cuyo funcionamiento está basado en la comparativa estructural de imágenes, y que puede aplicarse a nuevos dispositivos móviles de pequeño tamaño. El método que se ha empleado para la detección de esta tesis consiste en la aplicación de una medida de calidad de imagen, la similitud estructural, que permite determinar las diferencias existentes entre dos imágenes. La creciente necesidad de seguridad ha propiciado el desarrollo en los últimos años de muy variados sistemas de vigilancia. Es importante resaltar todos los avances que se han llevado a cabo en el campo del procesado de señales para hacer más eficiente a los sistemas de seguridad para esta tesis.

El estudio referente a este proyecto permitió ver aplicación de detección de movimiento cuyo funcionamiento está basado en la comparativa estructural de imágenes, y que puede aplicarse a nuevos dispositivos móviles de pequeño tamaño. Se observa como aplicaron métodos para la detección, que consisten en la aplicación de una medida de calidad de imagen, la similitud estructural, que permite determinar las diferencias existentes entre dos imágenes. Se revisa el análisis realizado a las diversas pruebas para estudiar el buen funcionamiento del método desarrollado para el proyecto mencionado.

En el documento de grado “Diseño y desarrollo de un dispositivo objeto controlado a través del sensor Kinect y la plataforma Arduino orientado al uso ludicopedagogico

de niños en la primera infancia" [4], el estudio de este proyecto, nos permitió observar cómo se hizo el diseño y desarrollo de un títere que detecta los movimientos corporales de las extremidades superiores e inferiores de las personas que se ubiquen en frente de él controlado a través de un sensor Kinect y la placa Arduino orientado en el uso lúdico - pedagógico de niños de la primera infancia. Se han analizado las diferentes teorías del aprendizaje que intentan dotar la enseñanza de aspectos útiles con el objetivo de tener una mejor perspectiva. Las teorías más influyentes estudiadas han sido el conductismo, el cognitivismo, el constructivismo y el conectivismo. Se estudia el sistema de capas que, con los planes de estudio actuales, partiendo desde primaria hasta los ciclos formativos de grado superior, permita emplear la robótica en todos los niveles educativos. Con estas premisas y enfocado en el construccionismo, una variante del constructivismo se el desarrollador de este proyecto logra estudiar el comportamiento tanto de la plataforma robótica, como la electrónica y programación a partir de unos patrones usando material asequible y normalizado. También se analiza el uso de los patrones tiene como objetivo facilitar la versatilidad, el modularidad, la interconexión entre módulos y reducir las variables que intervienen en el sistema. Se estudia la implementación de la tarea principal, que se encarga de detectar y seguir una línea negra sobre fondo blanco de modo autónomo. Cabe resaltar que el sistema microcontrolador implementado en el proyecto mencionado puede utilizarse en otras aplicaciones, aunque no sean robóticas, para conseguir el aprendizaje necesario en la tecnología de los microcontroladores y en su programación.

Con el desarrollo de un prototipo de captura de movimiento del cuerpo humano modelado en 3D y utilizando equipos de bajo costo, se obtiene la esqueletización de toda la masa en mención con cada uno de los ángulos detectados y su respectiva posición, y de esta manera predecir los movimientos y la capacidad de trabajar en conjunto con la robótica colaborativa, para evitar colisiones y accidentes de trabajo entre máquina y hombre en el macroproyecto de algoritmo de fusión de datos para prototipo de sistema de captura de movimiento en 3D utilizando la librería de venect.

1.7. Marcos de referencia

En esta sección del trabajo veremos algunos términos y definiciones que nos ayudaran a resolver con mayor facilidad nuestro trabajo.

1.7.1. Marco teórico

A través del tiempo, el campo de robótica enfoca precisamente a lo que uno conoce como Inteligencia artificial (IA) Controles difusos, Cobots (Robot Colaborativo), Domótica, IoT (internet of things), Sistemas Embebidos, Sistemas de captura, entre otros. Con base a esto, se busca como soporte de acercamiento en nuestro proyecto señalar los conceptos y estrategias respecto a nuestro tema de investigación.

De acuerdo con la revista internacional de tecnología, ciencia y sociedad: el sistema óseo del ser humano es complejo y para realizar el análisis, modelado y representación de los movimientos es una tarea ardua, sin embargo, con el desarrollo de las tecnologías de la información que permiten capturar las trayectorias de movimiento, ahorrando tiempo y esfuerzo para el diseño y desarrollo de herramientas de apoyo para áreas como la fisioterapia.

Captura de movimiento

Los sistemas de captura de movimiento son, por definición, sistemas que generan para un computador información que representan medidas físicas de movimiento capturado [5]. Consiste en vestir a un actor con un traje especial donde serán posicionados reflectores (Sistemas Ópticos) o transmisores (Sistemas Magnéticos) o básicamente sincronizar los movimientos físicos de algún miembro. Estos marcadores son generalmente posicionados en las llamadas “Articulaciones Universales”[6], que son 19 posiciones que ofrecen un mínimo de precisión para representación de un movimiento humano. En general los sistemas de captura tienen una estructura como muestra la Ilustración 1.

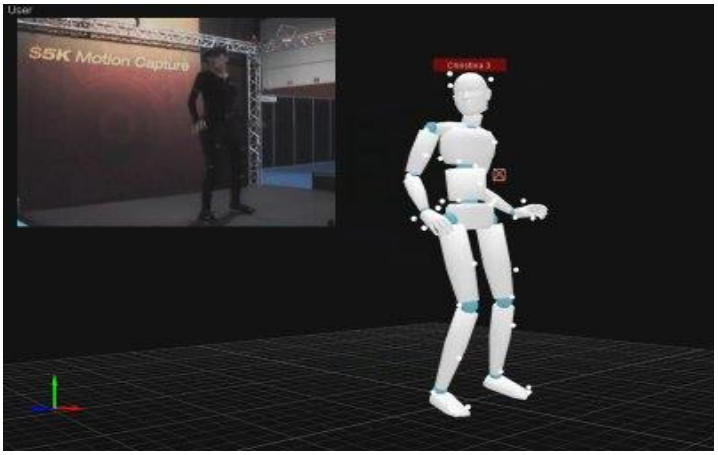

Ilustración 1 Sistema de captura de movimiento, tomada [6]

Visión Artificial

La visión artificial abarca en todas las aplicaciones industriales y no industriales en las que una combinación de hardware y software brinda un guiado operativo a los dispositivos en la ejecución de sus funciones de acuerdo con la captación y procesamiento de imágenes. cuentan con sensores digitales protegidos en el interior de cámaras industriales con ópticas especializadas para adquirir imágenes, de forma que el hardware y software informático pueden procesar, analizar y medir diversas características a la hora de tomar decisiones[8].

lustración 2 Visión artificial. tomada de [8]

La visión artificial la componen un conjunto de procesos destinados para realizar el análisis de imágenes que permita la obtención, procesamiento y análisis de cualquier tipo de información especial obtenida a través de imágenes digitales.

La visión artificial la componen un conjunto de procesos destinados para realizar el análisis de imágenes que permita la obtención, procesamiento y análisis de cualquier tipo de información especial obtenida a través de imágenes digitales.

Estos procesos son: captación de imágenes, memorización de la información, procesado e interpretación de resultados. También tiene aplicaciones en la industria actual como:

- Identificación e inspección de objetos.
- Determinación de la posición de objetos en el espacio.
- Mediciones Tridimensionales.
- Establecimiento de relaciones espaciales entre varios objetos (Guiado de robots).

Identificación artificial

Un sistema de reconocimiento óptico de caracteres (OCR) lee caracteres alfanuméricos sin conocimiento previo, mientras que un sistema de comprobación óptica de caracteres (OCV) confirma la presencia de una cadena de caracteres. Además, los sistemas de visión artificial pueden identificar piezas mediante la localización de un patrón único o identificar elementos según su color, forma o tamaño. Estas aplicaciones como DPM marcan un código o una cadena de caracteres directamente en la pieza. Fabricantes de todas las industrias suelen usar esta técnica para detectar errores, habilitar estrategias eficaces de contención y supervisar el control de procesos y las métricas de control de calidad [9].

1.7.2. Marco conceptual

- **Ruido:** Usualmente se encuentran dos tipos de ruido luminancia, causado por la falta de luz y el ruido de color, causado por el calentamiento de sensores. El ruido aparece cuando el color y la luz de algunos de los pixeles se ven alterados[10].

- **Cobots:** Un "Cobots" es un dispositivo robótico que manipula objetos en colaboración con un operador humano[11]. Un Cobots proporciona asistencia

al operador humano mediante la configuración de superficies virtuales que se pueden utilizar para restringir y guiar el movimiento. Si bien las pantallas hápticas servoactuadas convencionales también se pueden usar de esta manera, una distinción importante es que, mientras que las pantallas hápticas son dispositivos activos que pueden suministrar energía al operador humano, los Cobots son intrínsecamente pasivos. Esto se debe a que los Cobots no usan servos para implementar restricciones, sino que emplean juntas no holonómicas "dirigibles". Como consecuencia de su pasividad, los Cobots están potencialmente bien equipados para tareas críticas para la seguridad (por ejemplo, cirugía) o aquellas que involucran grandes fuerzas de interacción (por ejemplo, ensamblaje de automóviles).

- **Internet de las cosas (internet of things IoT):** El Internet de las cosas está impulsado por una expansión de Internet a través de la inclusión de objetos físicos combinados con la capacidad de proporcionar servicios más inteligentes al entorno a medida que haya más datos disponibles. Varios dominios de aplicaciones que van desde Green-IT y eficiencia energética a logística ya están empezando a beneficiarse de los conceptos de Internet de las cosas. Existen desafíos asociados con la Internet de las cosas, más explícitamente en las áreas de confianza y seguridad, la estandarización y la gobernabilidad requeridas para garantizar una Internet abierta y justa y confiable de las cosas que proporcione valor a toda la sociedad. El Internet de las cosas ocupa un lugar destacado en la agenda de investigación de varias multinacionales, así como de la Comisión Europea y países en pro de desarrollo como Colombia, en la ilustración 3 podemos visualizar como la inteligencia artificial enlaza los diferentes procesos de la vida diaria.

Ilustración 3 Internet of Things IoT [12]

- **Esqueletización:** La esqueletización busca representar una imagen binarizada (formada por pixeles blancos y negros) por un grafo de grosor fino (típicamente 1 pixel de grosor) cuyos puntos cumplen que la distancia local respecto a los bordes de la imagen binarizada es máxima. Los algoritmos de esqueletización se basan en la ejecución de un conjunto de iteraciones, donde en cada una se realiza el borrado de los pixeles pertenecientes a los bordes de la imagen, hasta que solamente queda el esqueleto[13].

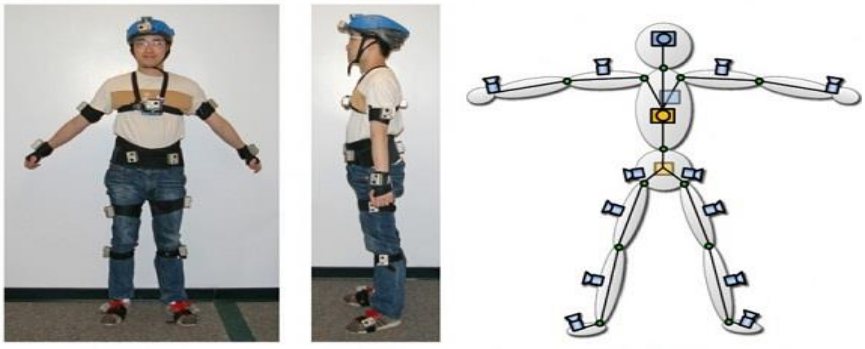

Ilustración 4 Sistema de esqueletización, tomada de [13]

- **Chroma Key:** Es un método para el recorte de una filmación, donde el fondo es de color azul o verde como lo muestra la ilustración cinco, se utilizan desde el inicio de la Televisión en aplicaciones simples hasta los sofisticados

sistemas de múltiples capas, básicamente consisten en insertar letras o imágenes en un vídeo, reemplazando un color por el video a insertar[14].

Ilustración 5 Chroma Key. Tomada de [14]

1.7.3. Marco legal y aspectos éticos

Las pruebas serán aplicadas al autor principales de este proyecto y seleccionando aleatoriamente a otras personas interesadas en participar, dando consentimiento y aprobación para la utilización de marcadores en las articulaciones, los cuales son círculos de foamy los cuales no son invasivos para el cuerpo humano, ni generan ningún tipo de repercusión. Es importante mencionar que este proyecto no contamina el ambiente ya que no genera desechos tóxicos, ni es necesario utilizar materia prima donde este se vea afectado.

El desarrollo del algoritmo para la captura de movimiento está enmarcado a la norma ISO / TS 15066: 2016, que especifica los requisitos de seguridad para los sistemas de robot industrial de colaboración y el entorno de trabajo, y complementa los requisitos y la guía sobre la operación de robot industrial de colaboración que figuran en ISO 10218-1 (Robots y dispositivos robóticos. Requisitos de seguridad para robots industriales - parte 1) e ISO 10218-2 (Robots y dispositivos robóticos. Requisitos de seguridad para robots industriales - parte 2). Se aplica a los sistemas de robots industriales como se describe en ISO 10218-1 e ISO 10218-2. No se aplica a robots no industriales, aunque los principios de seguridad presentados pueden ser útiles para otras áreas de la robótica [15].

1.8. Metodología

1.8.1. Línea de investigación

El proyecto se enfoca en dos líneas de investigación de la escuela de ingeniería de sistemas de la universidad del Sinú Seccional Cartagena, las cuales son: la línea automatización y robótica y la línea de inteligencia artificial del semillero de investigación DEARTICA.

1.8.2. Tipo de Investigación

Este proyecto esta enfocados a investigación aplicada, dado que cuenta con una problemática establecida, pero es necesario realizar una investigación para darle solución. Por este motivo con la adquisición de nuevos conocimientos respecto a la programación y desarrollo de algoritmos en Python y utilizando el algoritmo VNECT de C++ para la captura de movimiento, utilizando técnicas de visión por computador y demás temas que se desprenden, se logra establecer la esqueletización de la extremidad superior del cuerpo humano basado en video.

1.8.3. Definición de la Metodología

Este proyecto, está basado en el modelo de Martin y McClure y del SEI, que consiste en tres grandes fases [16]:

- Investigación y desarrollo inicial.
- Investigación aplicada.
- Transferencia.

A continuación, explicaremos como cada una de las fases se involucran en el desarrollo de nuestro proyecto.

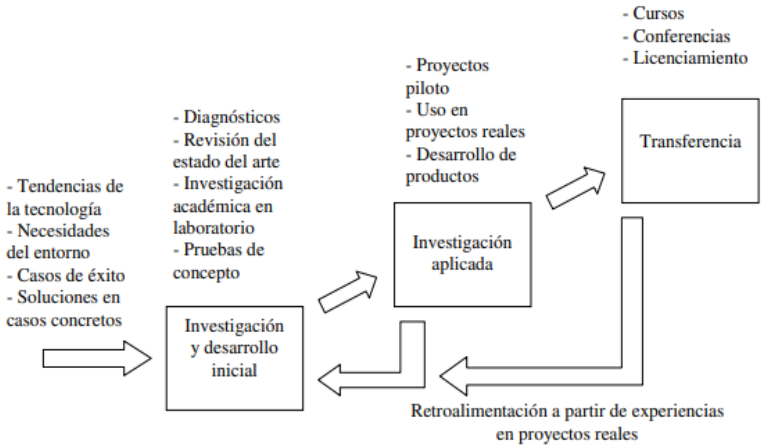

Ilustración 6 Esquema de la metodología. Tomada de [16]

- **Investigación y desarrollo inicial**

En la fase de análisis se realiza el análisis de los requerimientos del sistema de visión por computador.

En la fase de diseño y arquitectura se realiza el diseño hardware y software de los diferentes subsistemas de captura de movimiento. Para el diseño hardware se utiliza una metodología *Top-Down*, partiendo desde un diseño macro hasta llegar al nivel de detalle que permita la construcción del prototipo. Para el diseño software, se definirá una arquitectura modular y a partir de ella se desarrollarán cada uno de los módulos, que serán especificados en esta fase.

- **Investigación aplicada**

En la fase de desarrollo se trabajará utilizando lenguajes de programación de alto nivel como Python, y librerías especializadas como Open CV y *Tensor Flow*. Así mismo se implementarán los módulos hardware y software diseñados en la etapa previa, y obtenidos a través del kit de visión, pudiendo realizar pruebas parciales de la respuesta de cada módulo.

En la fase de pruebas se pueden ver como los procesos que permiten verificar la calidad del producto, es fundamental iniciar también con el proceso de identificación de fallos de sintaxis, implementación o usabilidad. Las pruebas son esenciales para asegurarse que las sentencias del código de programación del software sean correctas, y garantizar que la entrada definida produzca los resultados esperados.

- **Transferencia**

Se realiza la sustentación del proyecto en la Universidad del Sinú seccional Cartagena por parte de los autores principales, con la finalidad de ser evaluado y aprobado como proyecto de grado para la obtención del título de ingeniero de sistemas y se entregara el documento, junto a un artículo científico referente al proyecto.

2. DESARROLLO

En este capítulo se describe la funcionalidad de la AIY VISIÓN KIT de Google y todas sus fases, también se describen las técnicas, métodos y algoritmos que permiten lograr el propósito de este trabajo.

2. 1. Investigación, selección y pruebas de las técnicas de captura de movimiento

Para iniciar con el desarrollo de este trabajo, es necesario conocer las 2 maneras de ejecuta y poner a funcionar la cámara. La primera es usando la consola *secure Shell* y la segunda opción es usando la misma cámara como un ordenador.

2. 1. 1. Opción 1: Usar la consola *secure* Shell

Lo primero que se requiere para ejecutar y poder trabajar desde la consola secure Shell, es descarga la aplicación AIY Projects, para esto se debe dirigir a Google Play Store desde un teléfono, tal como se visualiza en la ilustración. https://play.google.com/store/apps/details?id=com.google.android.apps.aiy.[18]

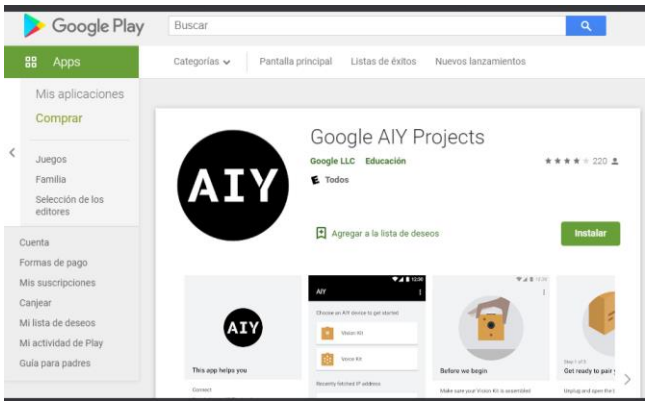

Ilustración 7 Aplicación GOOGLE PLAY. Tomada de [18]

Es de aclarar que para poder ejecutar la cámara con esta opción si el usuario tiene acceso a un teléfono inteligente Android y una computadora separada, para el desarrollo y la ejecución de la cámara se necesitarán los siguientes implementos:

- Teléfono inteligente Android.

- Computadora Windows, Mac o Linux.
- Conexión wifi.

Esta aplicación solo permitirá conocer la dirección ip de nuestro dispositivo (cámara) que a su vez se usará para comunicar el Kit de visión de forma inalámbrica a través de una computadora y SSH por separado, es importante aclarar que la computadora que se valla a usar esté en la misma red Wi-Fi que el kit de visión. Esto permitirá que el kit se conecte sin ningún problema a través de SSH.

Una vez que ya se tenga instalada la aplicación de **AIY PROYEC** y la consola ***Secure Shell*** en el computador lo siguiente es empalmar todo esto en un solo ordenador, Si está usando Chrome en una computadora con Windows, Mac o Linux, se deberá visualizar el ícono de *Secure Shell Extension* en la barra de herramientas. Se deberá hacer clic en ese icono y luego seleccione Diálogo de conexión en el menú que aparece.

Ilustración 8 Consola Secure Shell. Tomada [19]

Para sincronizar el kit de Visión es necesario que, en el campo superior, escriba pi@ + la dirección IP real de la Raspberry Pi. Después de escribir esto, podemos hacer clic en el campo del puerto. El botón [ENTER] Connect debería iluminarse. Haga clic en [ENTRAR] Conectar para continuar.

Ilustración 9 Conectar Consola. (fuente propia)

Muchas veces cuando se empieza a trabajar por primera vez con la consola *ssh(***o Secure Shell es un protocolo de comunicaciones seguras para conectar de forma remota dos sistemas operativos para que podamos controlar mediante consola de comandos un equipo host desde un equipo cliente**[20]*)*, por lo general hay que otorgarle ciertos permisos, es por eso que cada vez que esta lo requiera hay que otorgárselos y se pueda acceder a computadoras remotas como su Raspberry Pi, todos estos permisos son por una única vez. Ya habiendo otorgado los permisos necesarios para el uso de la consola en el computador hace falta uno de los más importante, es el permiso para trabajar en la consola, por lo general está siempre suele enviar un mensaje en ingles diciendo que, si el usuario quiere seguir con la conexión, así tal cual no los muestra la imagen

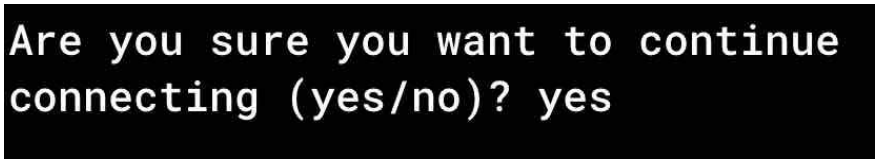

Ilustración 10 mensaje de aceptación. (fuente propia)

El siguiente punto en esta primera opción es colocar la contraseña de nuestra *Raspberry* pi, que por lo general siempre la contraseña que este dispositivo trae es

raspberry, hay que tener mucho cuidado al digitar esta contraseña, ya que al escribirla no se va a alcanzar a apreciar ninguna escritura y es un sistema muy sensible a mayúsculas y a minúsculas.

Al verificar que todos los permisos fueron otorgados satisfactoriamente y la contraseña fue digitada de manera exitosa, ya solo tocara comenzar a trabajar en nuestro entorno de trabajo que se indica en color verde y que comienza de la siguiente manera "**depi@raspberrypi: ~ $**", tal como no lo indica la imagen a continuación, pero antes de estos habrá una advertencia que indicara que la contraseña para el usuario de Raspberry Pi está configurada por defecto.

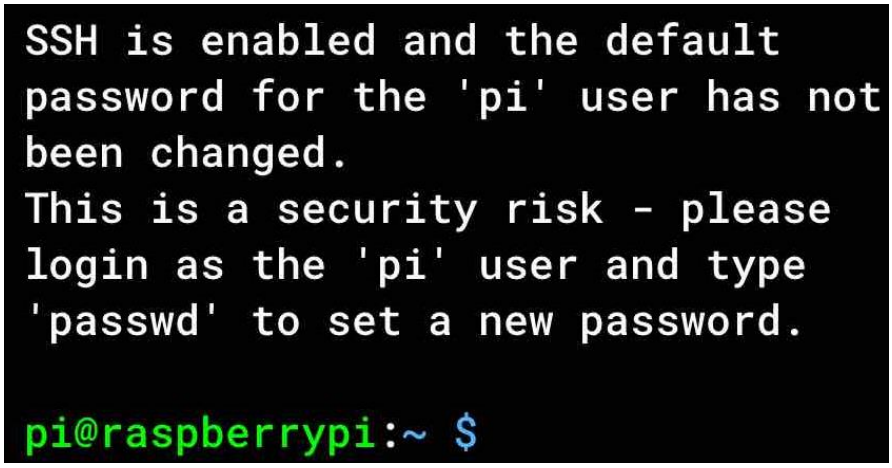

Ilustración 11 consola Activada. (fuente propia)

Ya con todo lo anteriormente explicado es como se podría trabajar con la **opción 1,** pero no se nos debe olvidar que también existe otra forma de trabajar con nuestro kit de Visión.

2. 1. 2. Opción 2: usar la misma cámara como un ordenador.

Elija esta opción si no tiene acceso a un teléfono inteligente Android o simple y llana mente si le gusta trabajar de forma más un grafica.

Para esta opción se Necesitarás:

- Computadora Windows, Mac o Linux

- Ratón (mouse).
- Teclado
- Monitor o TV
- Cable HDMI de tamaño normal y adaptador mini HDMI
- Adaptador para conectar un mouse y teclado al kit.

Esta opción es una de las más fáciles de configurar y de ensamblar, porque solo es conectar cada uno de los periféricos que se indica de la parte de atrás del kit de Visión y listo, hay que tener presente que para el teclado y mouse solo hay un solo puerto OTG, para poder conectar a este puerto hay que conseguir un cable otg con doble salida USB (mirar ilustración 12), para que de esta forma no halla problema alguno a la hora de las conexiones, la conexión debe quedar similar que la ilustración 13.

Ilustración 12 cable OTG. [21]

Ilustración 13 Conexión de periféricos. [22]

Dentro de este orden de ideas, al conectar todos los periféricos y esta no detecte ningún problema solo le quedara al usuario encender la cámara para que luego espere que esta inicie. Por consiguiente, para confirmar que esté conectado a la alimentación eléctrica, se podrá observar en el orificio de la tarjeta SD etiquetada en el cartón, claramente se apreciara un LED verde parpadeando en la placa Raspberry Pi, y también se visualizara el logotipo de Raspberry Pi en la esquina superior izquierda del monitor. El usuario tendrá que esperar unos segundos a que el dispositivo arranque correctamente y el LED deje de parpadear, lo que tomará aproximadamente dos minutos. Como resultado, se logrará visualizar que la pantalla del nuevo ordenador se tornará de color negro mientras se inicia, lo cual se recomienda tener un poco de paciencia mientras que el sistema carga correctamente y sin anomalías, al final el sistema quedará optimo cuando se escruche un pitido por parte de la cámara.

Luego de que ya todo se haya cargado satisfactoriamente se evidenciará el nuevo escritorio de trabajo, tal como lo muestra la siguiente ilustración

Ilustración 14 Escritorio de nuestro dispositivo. (fuente propia)

aquí ya se podrá comenzar a trabajar solo es abrir la terminar de trabajo para comenzar a crear nuevos proyectos con la cámara de visión. Aquí como en la opción

1 para iniciar a trabajar hay que colocar la contraseña de acceso de nuestra Raspberry pi, la cual es **raspberry**.

Ilustración 15 Consola desde la cámara. (fuente propia)

2. 2 Desarrollo de algoritmos

Considerando el conocimiento de las dos formas de trabajo de la cámara de kit visión, es necesario abordar el desarrollo de cada una de las opciones de trabajo que esta ofrece, como lo es la captura del rostro humano y la identificación de objetos entre otras más funciones. En este punto es inevitable recordar que el desarrollo de cada uno de los algoritmos está construido con el leguaje de desarrollo PHYTON.

Por consiguiente, para poder comprender y analizar donde se encuentra ubicado el algoritmo que ayuda al usuario a desarrollar las actividades de navegación en las diferentes librerías que ya se encuentran instalada en el sistema operativo de la cámara, esta se puede observar de la siguiente manera, ver ilustración 16:

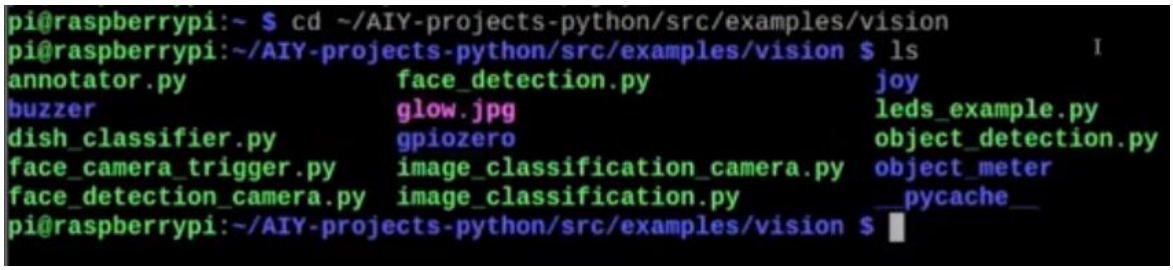

Ilustración 16 carpetas de toda la cámara. (fuente propia)

Teniendo en cuenta, lo anteriormente planteado, se puede deducir que los dos algoritmos se encuentran desarrollado de una manera única, por ende, las 2 funciones fueron diseñadas con el objetivo de visualizar estos pasos lógicos como si fuera una solo, estos serían los comandos para poder visualizar el código fuente de las funciones, ver ilustración 17.

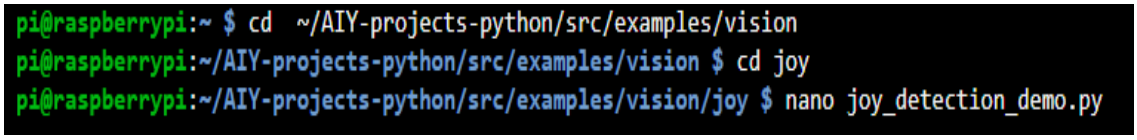

Ilustración 17 comando para visualizar código fuente. (fuente propia)

3 RESULTADOS

La puesta en ejecución de la cámara de AIY VISION KIT DE GOOGLE es todo un mundo entero por descubrir, esto se da gracias a la gran cantidad de modalidades de uso. En esta parte del proyecto como resultado se logra desarrollar y poner en práctica la inteligencia artificial de una manera sencilla para todo público, todo esto enfocado en cada uno de los objetivos y metodología de la investigación, que es captar en una forma fácil y sencilla los diferentes objetos dentro de un entorno o los rostros de una persona no importando el ambiente en que este se encuentre.

Para comenzar a visualizar y comenzar a trabajar con la cámara, es necesario detener la demostración de reconocimiento de rostro que viene activada por defecto, ya que esto es un gran impedimento a la hora de ejecutar cualquier otro comando o cualquier otra función, ya seguidamente cuando se requiera trabajar con función de captura de rostro se hará de manera manual, esto para tener la seguridad de que no representara ningún inconveniente para la investigación.

Para detener la demostración de captura de rostro automática se debe ejecutar el siguiente comando:

:~ $ Sudo systemctl stop joy_detection_demo

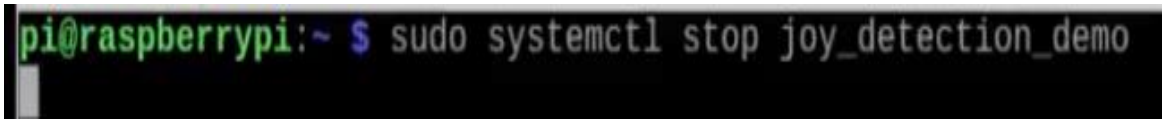

Ilustración 18 código para detener la cámara y comenzar a trabajar. (fuente propia)

Ejecutando el anterior comando ya se puede comenzar a trabajar sin ningún inconveniente.

Opción detección de rostro

A continuación, se mostrarán algunas imágenes de como la cámara va capturando los rostros de una o varias personas pasando, para cada rostro detectado se tendrá un nombre por defecto con el que será guardado como una imagen y poseerá un nombre similar a image.jpg, la captura imprimirá información como el puntaje

facial. También crea una imagen en la ubicación de salida, que es una copia de la imagen que incluye un cuadro alrededor de cada cara.

esta función se podrá poner en ejecución desde la consola el comando.**/face_detection_camera.py,** tal y como se muestra en la ilustración 19

```
pi@raspberrypi:~/AIY-projects-python/src/examples/vision $ ./face_detection_camera.py
```

Ilustración 19 código para activar función detección de rostro. (fuente propia)

Inmediatamente se mostrará un ejemplo de cómo la cámara va capturando el rostro de la persona encerrando cada rostro enfocado en un recuadro, cabe aclara que el rostro debe de estar descubierto, ya que si encuentra un obstáculo puede presentar problema y no mostrar ningún resultado y lo clasificara como **num_face=0**

En la siguiente imagen se mostrará una imagen con 4 rostros, uno de ellos cubierto con unos lentes, y los otros 3 están descubiertos.

Ilustración 20 imagen de ejemplo 1. (fuente propia)

En esta siguiente imagen se puede ver la preparación y la ubicación de la cámara

Ilustración 21 posicionamiento de la cámara y rostros. (fuente propia)

En las siguiente dos imágenes se podrá observar como la cámara va capturando los rostros de las personas que aparecen en las imágenes, pero la cuarta persona no la reconoce por tener lentes puestos

Ilustración 22 captación de rostro 1. (Fuente propia)

Ilustración 23 capación de rostro 2. (fuente propia)

En esta sección de imágenes se podrá visualizar como la cámara enumera la cara detectada, asignándole un valor de 0 a 1, donde 1 es un valor verdadero, y 0 es un valor que indica ausencia de rostro.

```
pi@raspberrypi:~/AIY-projects-python/src/examples/vision $ ./face_detection_camera.py
Iteration #0: num_faces=1
Iteration #1: num_faces=1
Iteration #2: num_faces=1
Iteration #3: num_faces=1
Iteration #4: num_faces=1
Iteration #5: num_faces=0
Iteration #6: num_faces=0
Iteration #7: num_faces=0
Iteration #8: num_faces=0
Iteration #9: num_faces=0
Iteration #10: num_faces=1
Iteration #11: num_faces=1
Iteration #12: num_faces=1
Iteration #13: num_faces=1
Iteration #14: num_faces=1
Iteration #15: num_faces=1
Iteration #16: num_faces=1
Iteration #17: num_faces=1
Iteration #18: num_faces=1
Iteration #19: num_faces=1
Iteration #20: num_faces=1
Iteration #21: num_faces=1
Iteration #22: num_faces=1
Iteration #23: num_faces=1
Iteration #24: num_faces=1
```

Ilustración 24 procesamiento de rostros captados. (fuente propia)

Opción clasificación de objetos

La demostración de la cámara en la opción de clasificación de imágenes utiliza un modelo de detección de objetos para identificar objetos a la vista del kit de visión. Para iniciar esta opción se es necesario escribir el siguiente comando en la consola de la cámara:

./image_classification_camera.py

Este proceso frecuentemente presenta demoras significativas al momento de ejecutarse, cuando termina de cargar aparecerá automáticamente una ventana en el entorno de trabajo de la cámara.

A continuación, se muestran dos ejemplos en donde se evidencia la manera que la cámara capta y entrega los resultados al activar la opción de clasificación de objetos, cabe aclara en esta parte del trabajo, que el algoritmo está capacitado para capturar muchísimos más objetos, pero para efectos prácticos y demostración de algoritmo se tomaron 2 objetos nada más:

Ejemplo 1: Detección de la imagen de una Copa.

Ilustración 25 presentación de elemento 1 hacia la cámara. (fuente propia)

Ilustración 26 captura de elemento 1. (fuente propia)

En esta primera imagen es una demostración de cómo la cámara de AIY VIVION KIT enfoca al artefacto, los cuales empieza a reconocer. En esta segunda imagen se observa el marco demostrativo de la cámara que va captando la imagen y otorgando el valor a cada toma.

Ilustración 27 captura de elemento 1. (Fuente propia)

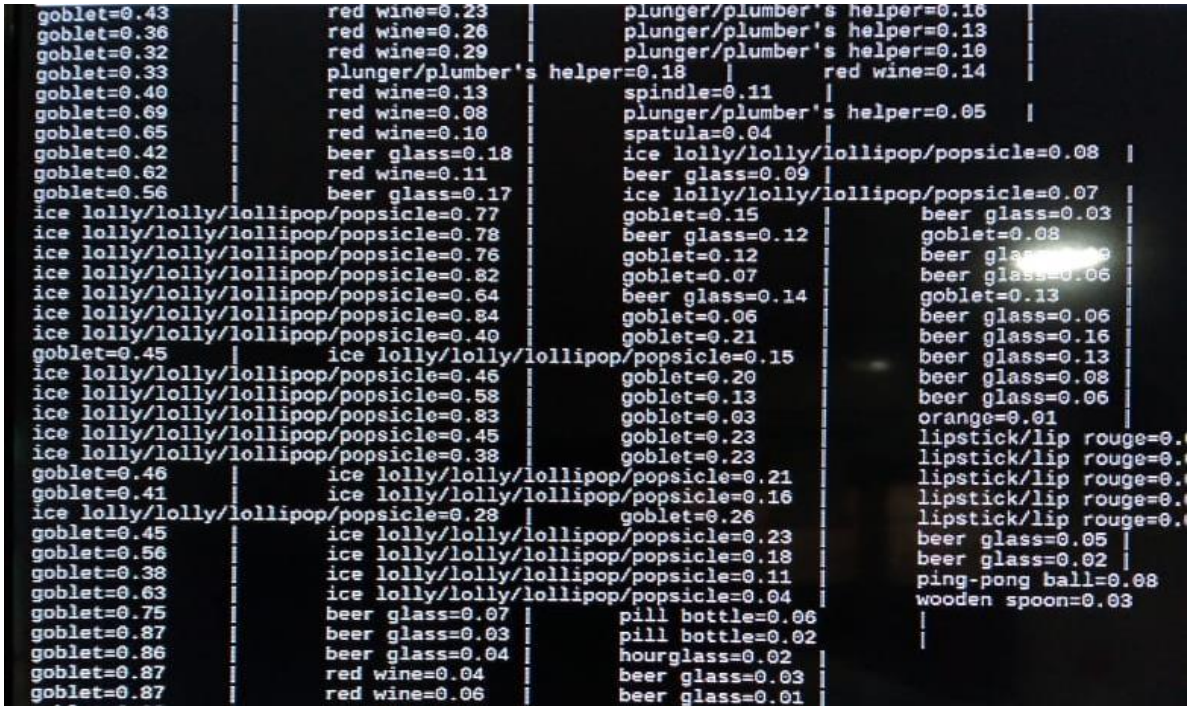

Ilustración 28 procesamiento y clasificación del objeto 1. (Fuente propia)

En este ejemplo se puede evidenciar como la cámara clasifica según diseño y tamaño del objeto enfocado, a pesar de que en el ejemplo se trabaja con una copa, la cámara los confunde con otros objetos similares, como lo son: un frasco, y le otorga un puntaje entre 0 y 0.40.

Por consiguiente, la cámara también tiende a confundir a la copa con una botella, el cual le otorga un puntaje que oscila entre 0 y 0.30. en este ejemplo con la copa también se puede visualizar otros objetos que se reconocieron como lo es también una cuchara, a la cual le otorga un puntaje de 0.21 y por último también se puede ver el valor que se le otorga al objeto original, el cual oscila entre 0 y 0.91.

En la siguiente tablan se mostrarán 5 tomas que se hicieron con la cámara desde diferentes distancias y sus respectivos resultados de acuerdo al objeto enfocado, en este caso sería una copa.

Tabla 1. Resultados de pruebas con una copa

Número de toma	Objeto enfocado	distancia	Resultado de la clasificación	Clasificación correcta
1	Copa	35 cm	• Copa 0.76% • Pelota de hielo 0.11% • Vino tinto 0.17% • frasco 0.03%	1
2	Copa	30 cm	• Copa 0.83% • Vaso de cerveza 0.25% • labios 0.60% • botella 0.10%	1
3	Copa	25 cm	• Copa 0.88% • frasco 0.30% • cuchara 0.21% • Bellota 0.12%	1
4	Copa	20 cm	• Copa 0.91% • Vino tinto 0.32% • frasco 0.40% • botella 0.30%	1
5	Copa	15 cm	• Copa 0.89% • Vaso de cerveza 0.70% • frasco 0.15% • Silla 70%	1

Ejemplo 2: Detección de la imagen de una raqueta.

Ilustración 29 presentación de elemento 2 hacia la cámara. (Fuente propia)

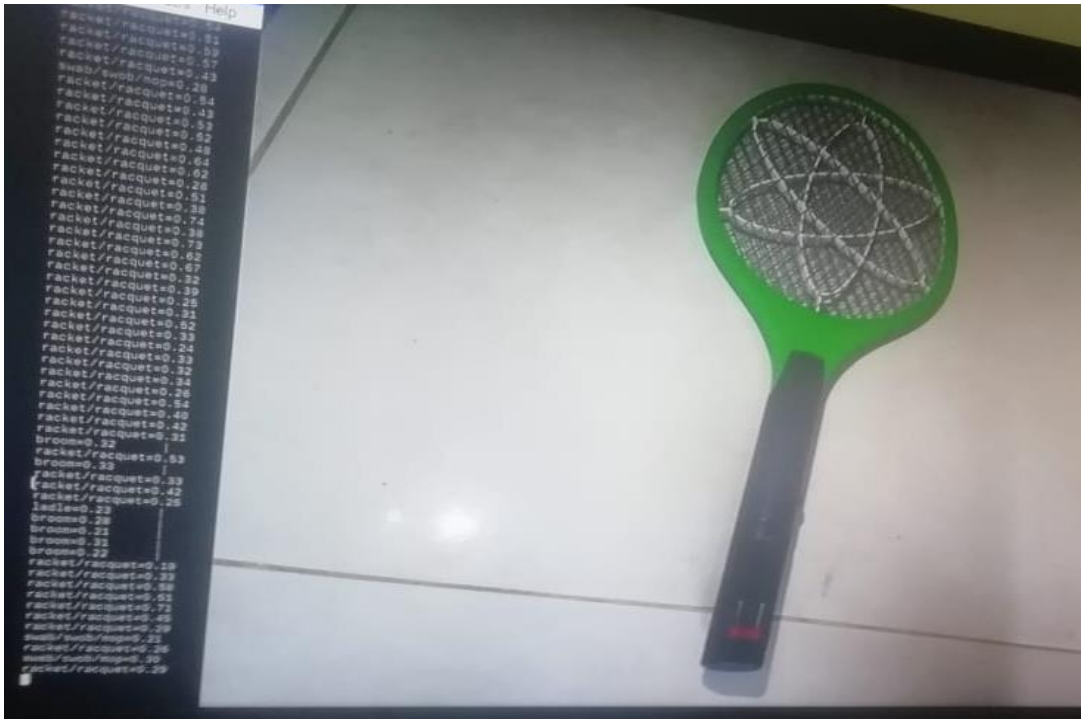

Ilustración 30 captura del elemento 2. (Fuente propia)

Dentro de este marco demostrativo, en esta segunda imagen se observa como la cámara va captando la imagen y el valor que le va otorgando a cada toma

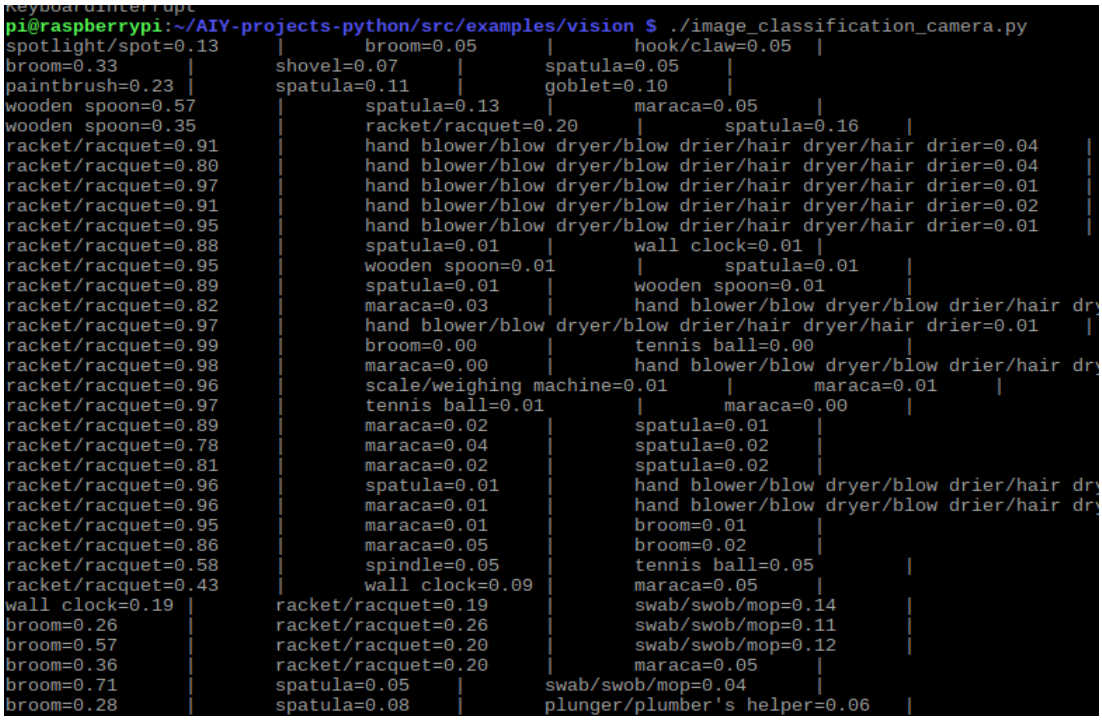

Ilustración 31 procesamiento y clasificación del objeto 2. (Fuente propia)

En este ejemplo 2 se puede evidenciar que la precisión de la cámara es más completa, esto se debe a que este implemento (raqueta) no guardas muchas similitudes con otras clases de objeto del mundo común, más sin embargo, no se puede dejar pasar por alto a los objetos que reconoce la cámara atreves de raqueta, entre los que podemos nombrar son: una maraca, la cual obtiene una puntuación que oscila entre 0.02 a 0.05, de igual forma se puede visualizar el reconocimiento de una espátula y la cámara le otorga una puntuación de 0.01. Pero a pesar de todo esto se observa que la cámara le da una mayor probabilidad a la raqueta, a la cual le concede una puntuación de un 0.98.

En la siguiente tablan se mostrarán 5 tomas que se hicieron con la cámara desde diferentes distancias y sus respectivos resultados de acuerdo al objeto enfocado, en este caso serían la raqueta.

Tabla 2. Resultados de pruebas con una raqueta

Número de toma	Objeto enfocado	distancia	Resultado de la clasificación	Clasificación correcta
1	Raqueta	40 cm	• Maraca 0.05% • Espátula 0.01% • Raqueta 0.95%	1
2	Raqueta	35 cm	• Estropajo(swob) 0.12% • Escoba(broom) 0.02% • Cuchara de madera 0.01% • Raqueta 0.95%	1
3	Raqueta	30 cm	• Maraca 0.10% • Raqueta 0.97% • Espátula 0.018% • Escoba(broom) 0.1%	1
4	Raqueta	25 cm	• Raqueta 0.98% • Maraca 0.8% • Espátula 0.15%	1
5	Raqueta	20 cm	• Raqueta 0.98% • Maraca 0.7% • Espátula 0.12%	1

CONCLUSIONES

Con la realización de este proyecto, se logró ampliar los cocimientos en diferentes áreas, como inteligencia artificial, robótica y las industrias 4.0, alcanzando un resultado positivo, a través de la evaluación funcional de la AIY de la visión de Google, es decir, fortalecimiento de los saberes previos, pero sobre todo tener una perspectiva más sencilla y clara de lo que es la visión artificial.

Adicionalmente se logró cumplir con el propósito general de la investigación que era desarrollar un prototipo de sistema de visión por computadora para apoyo en sistemas de robótica colaborativa mediante el kit AIY Visión de Google, con el fin de que este prototipo, sirva de base para futuras investigaciones, por consiguiente, ayudará a otros estudiantes de ingeniería a seguir fortaleciendo esta habilidad y así continuar mejorando dicha indagación.

Para ello, fue necesario diseñar una aplicación del sistema de visión por computadora para apoyo en sistemas de robótica colaborativa utilizando el kit AIY Google Visión, con esto se logró ensamblar la cámara de manera correcta, poniéndola en ejecución de una manera indicada, cumpliendo con los parámetros que se debe seguir para que no hubiera traumatismo más adelante y así se obtuviera un diseño óptimo.

Seguido de esto, se desarrolló una aplicación del sistema de visión por computadora para apoyo en sistemas de robótica colaborativa utilizando el kit AIY Google Visión. En esta etapa se pudo observar y analizar cada uno de los componentes de software que este tiene, logrando entender cada uno de los elementos.

Finalmente, para verificar el funcionamiento del sistema de visión por computadora como apoyo de los sistemas de robótica colaborativa, se realizó la etapa de pruebas, como resultado de ello se obtuvo la evaluación satisfactoria de la funcionalidad de la AIY Visión kit de Google. El algoritmo implementado permite la captura de rostros, con una efectividad de captura de un 100%, siempre y cuando el rostro a capturar este bajo luz controlada y no tenga ningún objeto que la esté

bloqueando, como por ejemplo unos lentes oscuros. Y la clasificación de objetos, los objetos probados con este algoritmo fueron: una copa y raqueta, la exactitud con que el modelo clasifica este tipo de objetos fue de un 91% para el caso de la copa y un 98% para el caso de la de la raqueta, luego de realizar 5 tomas para cada ejemplo.

En conclusión y en respuesta a la pregunta de investigación, para desarrollar el sistema de visión por computadora que permitan la integración sencilla con sistemas de robótica colaborativa es necesario adquirir y evaluar todo con referente al kit AIY visión de Google, ya que este kit es muy completo y sencillo de manejar para cualquier apasionado por la inteligenciar artificial y la visión por computadora

RECOMENDACIONES

Considerando la importancia que tiene esta investigación y en función de los resultados obtenidos se formulan algunas sugerencias para el personal que desee continuar con esta línea investigativa, esto con la finalidad de lograr adelantar lo más que se pueda y evitar retrasos innecesarios dentro del contexto de investigación, para ellos se hace énfasis en las siguientes recomendaciones.

Captura de rostro

- El candidato quien este de frente a la cámara debe tener la cara descubierta, principalmente los ojos
- Con las personas de test morena la cámara tiende a no capturar de manera correcta el rostro, esta debe estar quieta por cortos periodos de tiempo mientras se e identificada

Clasificación de objetos

- Al seleccionar el objeto a reconocer se recomienda revisar la compatibilidad técnica que tenga con la cámara, ya que hay muchos objetos con similitudes parecidas, esto conlleva a que la cámara se puede confundir con gran facilidad. Un objeto para tener en cuenta son los limones, con este ejemplo el porcentaje de clasificación puede ser 0 (cero) o null.

BIBLIOGRAFÍA.

[1] C. A. D. Celis y C. A. R. Molano, «Navegación de robot móvil usando Kinect, OpenCV y Arduino», *Rev. Prospect.*, vol. 10, n.º 1, pp. 71-78, 2012, doi: 10.15665/rp.v10i1.398.

[2] O. P. Osorio y F. L. Peña, «Captura de movimiento utilizando el Kinect para el control de una plataforma robótica controlada de forma remota por medio de seguimiento de los puntos de articulación del cuerpo», 2015, Accedido: sep. 17, 2019. [En línea]. Disponible en: http://repositorio.utp.edu.co/dspace/handle/11059/5136

[3] L. A. Blanquicett, M. C. Bonfante, y J. Acosta-Solano, «Prácticas de Pruebas desde la Industria de Software. La Plataforma ASISTO como Caso de Estudio», *Inf. Tecnológica*, vol. 29, n.º 1, pp. 11-18, feb. 2018, doi: 10.4067/S0718-07642018000100011.

[4] «[No title found]», *Tek. Rev. Científica*.

[5] «Revista Mexicana de Ingeniería Biomédica». http://www.rmib.mx/index.php/rmib (accedido sep. 21, 2019).

[6] «Software de captura de movimiento | Autodesk». https://latinoamerica.autodesk.com/solutions/motion-capture (accedido sep. 21, 2019).

[7] «AMR Producciones: CAPTURA DE MOVIMIENTO 3d». http://amrproducciones.blogspot.com/2011/07/captura-de-movimiento-3d.html (accedido oct. 08, 2019).

[8] «The page you are looking for can't be found | Cognex». https://www.cognex.com/esco/what-is/machine-vision/what-is-machine-vision?page=404 (accedido jun. 25, 2021).

[9] M. J. A. Ballesteros, «INTERFAZ GRÁFICA DE USUARIO PARA SIMULACIÓN Y DIDÁCTICA DE PROCESOS DE MOVIMIENTO EN ARTICULACIONES HUMANAS», p. 111.

[10] «Cómo evitar el ruido digital», *CasanovaFotoBlog*, jun. 26, 2014. https://www.casanovafoto.com/blog/2014/06/como-evitar-ruido-digital/ (accedido sep. 24, 2019).

[11] «Robotica i Programació: Fonts», *Robotica i Programació*. http://robotics-ugablog.blogspot.com/p/fonts.html (accedido sep. 24, 2019).

[12] «Internet industrial o industria 4.0 infográfico. Iconos infográficos de la industria 4.0 internet de las cosas red, solución | CanStock». https://www.canstockphoto.es/industrial-40-industria-infographic-50890050.html (accedido jun. 25, 2021).

[13] «Documento_por_capitulos%2F3_Capítulo_3.pdf». Accedido: jun. 25, 2021. [En línea]. Disponible en: http://bibing.us.es/proyectos/abreproy/12112/fichero/Documento_por_capitulos%252F3_Cap%C3%ADtulo_3.pdf

[14] «Cómo iluminar un Chroma Key de Forma Perfecta», *Agencia de Marketing Online Zaragoza*, abr. 26, 2018. https://www.popcornstudio.es/chroma-key (accedido jun. 25, 2021).

[15] 14: 00-17: 00, «ISO/TS 15066:2016», *ISO*. http://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/06/29/62996.html (accedido sep. 24, 2019).

[16] «MIFISIS2004.pdf». Accedido: sep. 24, 2019. [En línea]. Disponible en: https://web.imt-atlantique.fr/x-info/harbol07/MIFISIS2004.pdf

[17] «MIFISIS2004.pdf». Accedido: jun. 25, 2021. [En línea]. Disponible en: https://web.imt-atlantique.fr/x-info/harbol07/MIFISIS2004.pdf

[18] «Google AIY Projects - Apps en Google Play». https://play.google.com/store/apps/details?id=com.google.android.apps.aiy (accedido jun. 25, 2021).
[19] «Vision». https://aiyprojects.withgoogle.com/vision/ (accedido abr. 22, 2020).
[20] «▷ Como usar SSH en Windows 10», *Profesional Review*, nov. 30, 2018. https://www.profesionalreview.com/2018/11/30/ssh-windows-10/ (accedido may 22, 2020).
[21] «KUBII». https://www.kubii.es/recherche?controller=search&orderby=position&orderway=desc&search_query=otg+doble+usb&submit_search= (accedido abr. 22, 2020).
[22] «Vision», *Vision*. https://aiyprojects.withgoogle.com/ (accedido abr. 22, 2020).

ANEXO 1

Ya para terminar con la presentación de este trabajo de grado, se mostrará la codificación realizada para desarrollar las pruebas de evaluación de las 2 funciones trabajadas.

Código para detección de rostro.

```
import argparse
from picamera import PiCamera
from aiy.vision.inference import CameraInference
from aiy.vision.models import face_detection
from aiy.vision.annotator import Annotator
def avg_joy_score(faces):
    if faces:
        return sum (face.joy_score for face in faces) / len(faces)
    return 0.0
def main ():
    parser = argparse. ArgumentParser ()
    parser.add_argument ('--num_frames', '-n', type=int, dest='num_frames', default=None,
        help='Sets the number of frames to run for, otherwise runs forever.')
    args = parser. parse_args()
    # Forced sensor mode, 1640x1232, full FoV. See:
    # https://picamera.readthedocs.io/en/release-1.13/fov.html#sensor-modes
```

```
    # This is the resolution inference run on.
    with PiCamera (sensor_mode=4, resolution = (1640, 1232), framerate=30) as camera:
        camera. start_preview ()
        # Annotator renders in software so use a smaller size and scale results
        # for increased performace.
        annotator = Annotator (camera, dimensions=(320, 240))
        scale_x = 320 / 1640
        scale_y = 240 / 1232
        # Incoming boxes are of the form (x, y, width, height). Scale and
        # transform to the form (x1, y1, x2, y2).
        def transform(bounding_box):
            x, y, width, height = bounding_box
            return (scale_x * x, scale_y * y, scale_x * (x + width),
                    scale_y * (y + height))
        with CameraInference (face_detection. Mode l ()) as inference:
            for result in inference.run(args.num_frames):
                faces = face_detection.get_faces(result)
                annotator. clear ()
                for face in faces:
                    annotator. bounding_box (transform (face. bounding_box), fill=0)
                annotator. update ()
```

```
            print ('#%05d (%5.2f fps): num_faces=%d, avg_joy_score=%.2f' %
                (inference. count, inference.rate, len(faces), avg_joy_score(faces)))

        camera. stop_preview()

if __name__ == '__main__':
    main ()
```

Código para clasificación de objetos.

```
import argparse
import contextlib
from aiy. vision. inference import CameraInference
from aiy. vision. models import image_classification
from picamera import PiCamera

def classes_info(classes):
    return ', '. join ('%s (%.2f)' % pair for pair in classes)

@contextlib. contextmanager
def Camera Preview (camera, enabled):
    if enabled:
        camera. start_preview()
    try:
        yield
    finally:
        if enabled:
            camera. stop_preview()

def main ():
```

```
    parser = argparse. ArgumentParser ('Image classification camera
inference example.')
    parser.add_argument ('--num_frames', '-n', type=int, default=None,
        help='Sets the number of frames to run for, otherwise runs forever.')
    parser.add_argument ('--num_objects', '-c', type=int, default=3,
        help='Sets the number of object interences to print.')
    parser.add_argument ('--nopreview', dest='preview', action='store_false',
default=True,
        help='Enable camera preview')
    args = parser. parse_args()

    with PiCamera (sensor_mode=4, framerate=30) as camera, \
         Camera Preview (camera, enabled=args. preview), \
         CameraInference (image_classification. model ()) as inference:
        for result in inference.run(args.num_frames):
classes = image_classification.get_classes (result, top_k=args.num_objects)
            print(classes_info(classes))
            if classes:
                camera. annotate_text = '%s (%.2f)' % classes [0]

if __name__ == '__main__':
    Main ()
```