

UNIVERSIDAD NACIONAL DE INGENIERÍA

Recinto Universitario “Simón Bolívar”

FACULTAD DE INGENIERÍA DE ELECTROTECNIA Y COMPUTACION



Lider en Ciencia y Tecnologia

Trabajo Monográfico:

“Desarrollo de un sistema de reconocimiento de imágenes utilizando Raspberry Pi y MATLAB® que contribuya al aprendizaje del tratamiento digital de señales”

Autor:

Br. Pavel Fernando Avilés Andino

Tutor:

Ing. Álvaro Gaitán

Managua, Nicaragua Mayo 2016

AGRADECIMIENTO

El presente trabajo de tesis primeramente me gustaría agradecerle a Dios por haberme permitido llegar hasta aquí y por hacer realidad este sueño tan anhelado. A la Universidad Nacional de Ingeniería UNI Por darme la oportunidad de estudiar y ser un profesional. Y a todas esas personas que han formado parte de mi vida como estudiante me encantaría agradecerles por su apoyo, amistad, consejos, ánimos y compañía para ellos muchas gracias y que Dios los bendiga.

DEDICATORIA

A Dios, por haberme permitido llegar hasta este punto y haberme dado salud para lograr mis objetivos académicos y por su puesto a mi familia y mis padres que en todo momento recibí su apoyo para ellos esta dedicatoria de tesis.

RESUMEN

El presente documento presenta los pasos seguidos y resultados obtenidos en el diseño de medios de laboratorios enfocados en el procesamiento digital de imágenes para el aprendizaje. Otorgar a los estudiantes una herramienta para el entendimiento y comprensión de los distintos algoritmos utilizados en esta área.

INDICE

Introducción	1
Objetivos	2
Justificación	3
CAPITULO I. MARCO TEORICO	4
1.1 Raspberry Pi	5
1.1.1 Componentes Básicos	6
1.1.2 SoC (System on Chip)	11
1.1.3 Paquete de Soporte de Matlab y Simulink para Raspberry Pi	13
1.2 Introducción al Procesamiento Digital de Imágenes (PDI)	14
1.2.1 Definición	14
1.2.2 Áreas de Aplicación	15
1.2.3 Conceptos básicos de las imágenes en MATLAB®	18
1.2.4 Detección de Bordes (EDGE)	21
1.2.4.1 Operadores basados en la primera derivada (Gradiente).....	21
1.2.4.2 Operadores de Roberts	23
1.2.4.3 Operadores de Prewitt, Sobel y Frei-Chen	23
1.2.5 Speeded Up Robust Features (SURF)	25
1.2.5.1 Fundamento Teórico	27
1.2.6 Histogramas de Gradientes Orientados (HOG) y descriptores de HOG.....	32
1.2.7 Maquina de vectores de soporte (SVM)	36
1.3 Enfoque Constructivista	43
1.3.1 Características del aprendizaje constructivista	43
CAPITULO II. ANALISIS Y PRESENTACION DE LOS RESULTADOS	45
2.1 Diseño Metodológico	45
2.2 Análisis de la problemática	46
2.2.1 Encuestas aplicadas a los estudiantes que participaron como grupo de prueba para las guías desarrolladas	46
2.2.2 Entrevistas aplicadas a docentes	50
2.3 Diseño de guías de laboratorios	51
2.4 Implementación de las guías de laboratorio	53
2.4.1 Laboratorio 1: Captura y pre-procesamiento de las imágenes	54
2.4.2 Laboratorio 2: Detección de los bordes de una imagen	54

2.4.3 Laboratorio 3: Detección de Objetos a través de características	54
2.4.4 Laboratorio 4: Detección de rostros y cuerpos	55
2.5 Evaluación de resultados.....	56
2.5.1 Encuesta final aplicada a estudiantes con los que se implementaron las guías	56
2.6 Resultados de las encuestas finales aplicadas a los estudiantes	59
CAPITULO III. CONCLUSIONES Y RECOMENDACIONES	66
3.1 Conclusiones.....	66
3.2 Recomendaciones	67
REFERENCIAS	68
ANEXOS.....	69

INTRODUCCION

En este documento se describe el desarrollo del trabajo monográfico, los objetivos planteados, incluyen las herramientas teóricas que se necesitaron para la ejecución del proyecto, los resultados obtenidos a través de la aplicación de instrumentos para el proceso de desarrollo del trabajo en cuestión.

El trabajo monográfico consiste en el diseño e implementación de guías de laboratorios enfocadas al procesamiento digital de imágenes. En estas guías se abordaron actividades que demostraron los principios de algunos algoritmos utilizados en esta área. El conjunto de guías de laboratorios comprenden tres técnicas muy utilizadas.

Estas guías se diseñaron con el propósito de que los estudiantes obtengan un aprendizaje significativo de los contenidos abordados en cada una de las prácticas de laboratorio realizadas, permitiendo la ampliación de conocimientos y el desarrollo de destrezas necesarias para su campo laboral.

El desarrollo del trabajo monográfico se realizó con participación de 10 estudiantes de las carreras de Ingeniería Electrónica e Ingeniería en Telecomunicaciones. Primeramente, se aplicaron encuestas a los estudiantes para determinar el nivel de motivación y conocimientos que los estudiantes tenían. Los resultados de las encuestas se utilizaron como insumos para el diseño de las guías de laboratorios. Seguidamente, se implementaron las guías de laboratorio; en esta etapa se evaluó el desempeño de los estudiantes y mejoras de las guías en función del desempeño de ellos. Por último, se evaluó el logro obtenido de las prácticas de laboratorio mediante la aplicación de una encuesta final.

OBJETIVOS

OBJETIVO GENERAL

Desarrollar un prototipo de un sistema de visión artificial que contribuya a la enseñanza y adiestramiento de los estudiantes en el procesamiento y reconocimiento de imágenes utilizando la plataforma Raspberry Pi a través del software MATLAB®.

OBJETIVOS ESPECÍFICOS

1. Desarrollar experimentos para adquirir los conocimientos fundamentales que se emplean en el procesamiento digital de imágenes utilizando Raspberry Pi y MATLAB.
2. Diseñar medios de laboratorios desde un enfoque constructivista, para que los estudiantes puedan adquirir y procesar imágenes en tiempo real, aplicables a la asignatura de Tratamiento Digital de Señales.
3. Evaluar el contenido desarrollado por medio de rúbricas, encuestas y entrevistas con grupos muestrales, esto enfocado al aprendizaje significativo y constructivista.

JUSTIFICACIÓN

La Facultad de Electrotecnia y Computación (FEC), no posee equipos de procesamiento de imágenes, puesto que el equipo presente en el laboratorio de Automatización no es funcional, y solamente existe uno para aproximadamente 30 alumnos por grupo de clases. Esto representa una gran limitante para la adquisición de conocimientos y competencias tanto para docentes como estudiantes.

Con esta tesis se ayudara a mejorar estas limitaciones, proporcionando medios de laboratorio y una alternativa que les permita a los estudiantes conocer, comprender y entrenarse en temas de adquisición, procesamiento y análisis de imágenes, haciendo uso de tecnologías nuevas y de bajo costo.

El material desarrollado con esta presente tesis estará enfocado principalmente a la carrera de Ingeniería en Telecomunicaciones, específicamente a la rama de Control de Procesos, pero también se pueden desarrollar proyectos finales de cursos para las asignaturas como por ejemplo Electrónica Analógica II, Electrónica Digital II y Microprocesadores, brindando un gran beneficio para la carrera de Telecomunicaciones, cabe destacar que también la carrera de Ingeniería Electrónica se vería beneficiada, ya que en esta se imparten asignaturas similares en las cuales se puede aprovechar el material desarrollado.

CAPITULO I. MARCO TEÓRICO

1.1 Raspberry Pi

Raspberry Pi es un ordenador que tiene un tamaño tan pequeño como el de una tarjeta de crédito. Un grupo de Ingenieros y académicos de la Universidad de Cambridge, decidieron desarrollar un ordenador que todo el mundo pudiera permitirse el lujo de comprar para crear un ambiente de aprendizaje en la programación. El proyecto Raspberry Pi se convirtió en algo prometedor desde que empezaron a aparecer los procesadores móviles baratos y de gran alcance con muchas características avanzadas que permite un posible desarrollo de Raspberry, que se continuó bajo la Fundación Raspberry Pi con su primer lanzamiento en 2012 (Raspberry Pi Foundation, s.f.).



Figura 1.1: Raspberry Pi y Raspberry Pi 2 (Pi, Power, 2015)

Hasta la primavera del 2015, la Fundación Raspberry Pi ha lanzado cinco modelos en dos generaciones de la computadora. La primera generación compuesta de cuatro modelos como se muestra en la figura. Los dos últimos modelos son versiones mejoradas de sus versiones anteriores para que el equipo sea más eficiente y conveniente para el usuario, principalmente por tener menor consumo de energía y más puertos USB¹. La segunda generación se compone de un solo modelo llamado Raspberry Pi 2 modelo B, con un CPU más rápido y más memoria (Foundation, 2015).

¹ Universal Serial Bus

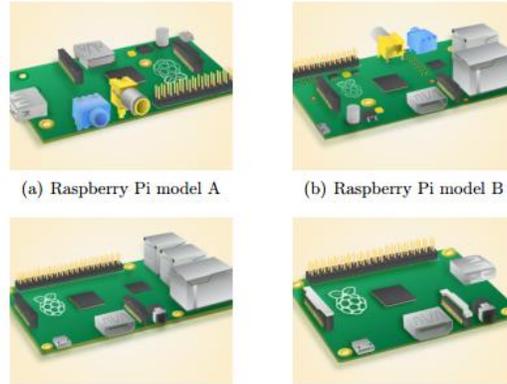


Figura 1.2: Modelos Raspberry Pi (RPiHardware, 2015)

A inicios del 2015, la segunda generación de computadoras Raspberry Pi fue lanzada, esta es llamada Raspberry Pi 2 Modelo B. Este modelo representa mejoras significativas con respecto a sus predecesores lo cual fue bien recibido por la comunidad que utiliza estos equipos. Raspberry Pi 2 modelo B utiliza un procesador BCM2836 mucho más potente que ofrece un CPU ARM Cortex A7 quadcore de 900MHz, 1 GB de RAM y las mismas especificaciones del procesador gráfico permitiéndole correr aproximadamente 6 veces más rápido en comparación con sus predecesores, a un precio sin cambios. Por otra parte, los soportes de la versión Windows 10 IoT² está disponible para este modelo de forma gratuita brindando un gran potencial para usos futuros.

En la siguiente tabla podemos ver un resumen de las principales características de cada modelo:

Tabla 1.1: Raspberry Pi, Modelos y Especificaciones

Raspberry Pi	Generación 1			Generación 2	
Especificaciones	Modelo A	Modelo B	Modelo A+	Modelo B+	Modelo 2 B
Alimentación	300mA	700mA	200mA	600mA	900mA
Puerto Ethernet	No	Si	No	Si	Si
Puertos USB	1	2	1	4	4
GPIO	26	26	40	40	40
Puerto SD	SD	SD	microSD	microSD	microSD
SoC	BCM2835	BCM2835	BCM2835	BCM2835	BCM2836

² Internet Of Things

900 mA, debido a que cuenta con un procesador de 4 núcleos. Se requiere mayor potencia para un funcionamiento más estable (Pi, Power, 2015).

- **Puerto HDMI (*High-Definition Multimedia Interface*):** es la conexión para pantalla de Alta Definición (HD) ya sean monitores o televisores LCD. Todos los modelos brindan una resolución desde 640x350 a 1920x1200, ya sea en los formatos estándares PAL o NTSC.
- **Conector para interfaz de cámara serial (*CSI*):** este es un conector para un módulo de cámara. La cámara es específicamente diseñada para la Raspberry Pi y puede ser conectada con un cable de cinta que brinda soporte tanto para fotografía como video de alta definición (Pi, Camera Module, 2015).
- **Conector de audio:** es una salida de audio a través de un Jack de 3.55 mm. En los modelos recientes funciona solo como salida de audio.
- **Puerto Ethernet:** es un conector de red que brinda velocidades de 10/100 Mbits/s a través de un cable RJ45. Este puerto está disponible en los modelos B, B+, y Raspberry 2 B.

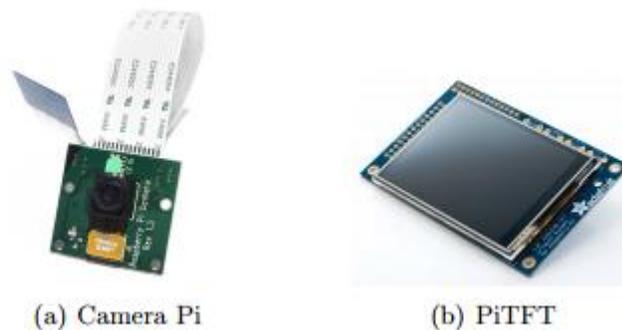


Figura 1.4: Cámara Pi y PiTFT

- **Puertos USB:** con este puerto tenemos compatibilidad con cualquier dispositivo como teclados, mouse, Wi-Fi dongle, o cámara WEB. La versión del USB es 2.0. Los modelos A y A+ tienen solo un puerto mientras que el modelo B tiene 2. En los modelos B+ y Raspberry Pi 2 B se incluyen 4.
- **GPIO:** es un set de pines de entrada y salida de propósito general, en los cuales podemos conectar distintos periféricos, por ejemplo tarjetas de

expansión, dispositivos de control del CPU. los modelos A y B tienen 24 pines presentes mientras que en los A+, B+ y la segunda generación tienen 40.

- **LED's:** sirven para indicar el estado de la tarjeta, por ejemplo, cuando está alimentada el LED rojo se enciende y si está conectada a una red el LED amarillo se activa.
- **Display Serial Interface (DSI):** este conector sirve para conectar un display, como el PiTFT, el cual es una pantalla de cristal líquido (Thin Film Transistor).
- **Conector SD-Card:** sirve para insertar una memoria SD que almacena el Sistema Operativo (OS). Raspberry Pi soporta hasta 32 GB de almacenamiento con memorias Clase 10 para altas velocidades. En los últimos modelos el puerto SD fue sustituido por un puerto para micro SD.

Una de las principales características son sus pines GPIO (General Purpose Input/Output). GPIO es un pin genérico y reusable el cual puede ser configurado con valores lógicos, 1 y 0, lo cual permite que pueda actuar como entrada o salida de acuerdo al propósito del usuario con el fin de indicar el estado de los dispositivos periféricos. Los pines GPIO se configuran de forma individual, sin embargo la configuración de pines GPIO como un grupo de 8 pines normalmente resulta en un puerto GPIO que permiten realizar las mismas operaciones individualmente como la lectura de entrada y escribir resultados de salida.

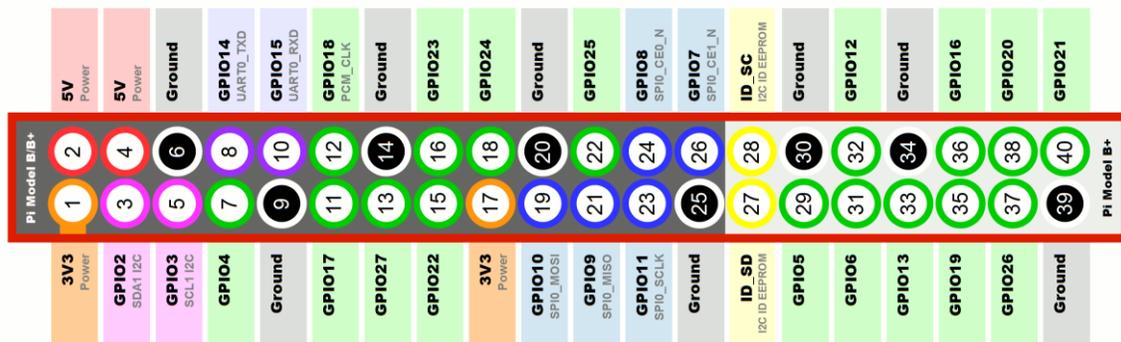


Figura 1.5: Diagrama de Pines GPIO

Los pines GPIO permiten control sobre algunas funciones tales como Comunicación Serial (UART), Bus de Interfaz de Periféricos Serial (SPI), Circuito Inter-Integrado (I2C), Hardware adjunto encima (HAT), acceso a la ROM programable y borrable eléctricamente (EEPROM) así como también nos proporciona 3.3 V y 5 V.

Los colores de los pines mostrados en el diagrama indican la funcionalidad la cual será explicada a continuación:

- **Fuente de Voltaje:** los pines de color naranja y rojo son los que brindan dos niveles de voltajes, 3.3 V y 5V, en el cual 3.3V los brindan los pines 1 y 17 los cuales son considerados como Fuentes de Alimentación mientras que 5V los tenemos en los pines 2 y 4 los cuales tienen como propósito interactuar con otros dispositivos que requieran 5V.
- **I2C:** pines de color rosa, son los pines 3 y 5 para Línea Serial de Datos (SDA) y Línea Serial de Reloj (SLC). I2C permite comunicación a corta distancia entre dispositivos en el mismo circuito, así como también el control de los dispositivos periféricos conectados a la tarjeta vía cable. Los modos de transferencia de datos disponibles son 100 kbp/s, 400 kbp/s, 1 Mbp/s, 3,4 Mbps/s y 5 Mbps/s. I2C en Raspberry Pi se puede utilizar para sensores de control u otras interfaces.
- **UART:** los pines en morado, el número ocho y el diez son para transmisión de datos (TxD) y recepción (RxD). UART es una comunicación serial que permite que dos dispositivos intercambien datos a través de sus interfaces seriales. La conexión es realizada de tal manera que el transmisor del primer dispositivo es conectado al receptor del segundo y viceversa, también sus tierras tienen que estar conectados. UART permite a la Raspberry Pi conectarse a una computadora, un microcontrolador y cualquier dispositivo con una interfaz serial.
- **SPI:** coloreado en azul, es una interfaz de comunicación realizada en forma de maestro y esclavo, el cual permite que un maestro controle uno o más esclavos a través de cuatro pines llamados Master Output Slave Input (MOSI), Master Input-Slave Output (MISO), Serial Clock (SCLK) y Slave

Select (SS) también conocido como Chip Enable (CE). Los datos son transmitidos en modo Full – Duplex el cual consiste en que el maestro envía los datos a través de la línea MOSI y los esclavos reciben los datos por la línea MISO correspondiente a una señal de reloj generada en SCK por un maestro. Raspberry Pi ofrece dos conexiones del tipo SPI, estas estas nombradas como SPI0 y SPI1. SPI0 está en los pines número 19, 21, 23, 24 y 26 permite el control sobre los esclavos mientras que SPI1 presente 11, 12, 35, 36, 38 y 40 permite el control sobre tres esclavos.



Figura 1.6: HAT con pines más largos

- **HAT:** pines en amarillo, se proporciona en el pin número 27 y 28 para el propósito de conectar tarjetas add-on Hat como se muestra en la Figura 1.6, que permite Raspberry Pi para configurar automáticamente pines GPIO y su driver correspondientes a un dispositivo conectado a través de I2C datos de la EEPROM y el reloj, a saber ID_SD y ID_SC respectivamente.

Por defecto todos los pines GPIO de las Raspberrys Pi actúan como entradas de propósito general a excepción de los pines asignados a Alimentación, Tierra (GND) y UART. Sin embargo, los pines UART pueden ser configurados como pines de propósito general. Para realizar cualquier operación utilizando los pines GPIO, los pines deben estar configurados para ser ambas cosas, sea entradas o salidas, por ejemplo, programando a través de Python o C o por toolbox de Matlab.

1.1.2 SoC (System on Chip)

El SoC de Raspberry Pi involucra el CPU, GPU y la RAM. El CPU es procesador basado en ARM el cual es uno de los CPU más comunes en los teléfonos móviles. La primera generación de Raspberry Pi utiliza un procesador ARM BCM2835 el cual ofrece una arquitectura versión 6. En la segunda generación, la tarjeta utiliza ARM BCM2836 versión 7 conocida como Cortex-A7 el cual soporta una variedad más amplia de software.

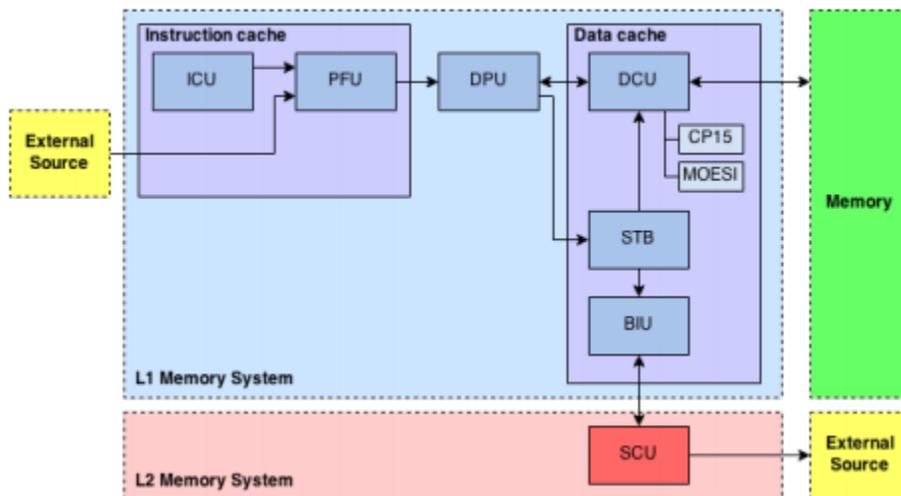


Figura 1.7: Diagrama de Bloques ARM Cortex - A7

El procesador gráfico proporcionado por Broadcom es un GPU de procesamiento gráfico 3D de doble núcleo, el cual tiene un rendimiento relativo a la Xbox 360 o el doble de desempeño que un iPhone4S. La GPU es capaz de decodificar y codificar video de calidad en formato H.264, 1080p Bluray y tiene una velocidad de 24 GFLOPs³ para operaciones de computo de propósito general paralelizadas como la multiplicación de matrices.

VideoCore IV GPU, como muestra en la figura 1.8, consta de shader o procesador sombreado de propósito general de punto flotante conocido como Quad Processing Unit (QPU). Cada QPU es una instrucción única de cuatro vías, procesador de multiples datos (SIMD) pero puede ser considerado un procesador de 16-way SIMD

³ Operación de coma flotante (gigaFLOPS)

desde que sus cuatro vectores reales de cuatro valores se ejecutan simultáneamente bajo una instrucción para tiempos de 4 ciclos de reloj en el cual cada tiempo es multiplexado, dando un vector de 16 valores después de completar los 4 ciclos. En otras palabras, una instrucción puede ser emitida a cada QPU cada cuatro ciclos de reloj, por lo tanto, la operación parece ser dirigida bajo un único ciclo que da un valor de salida de 16.

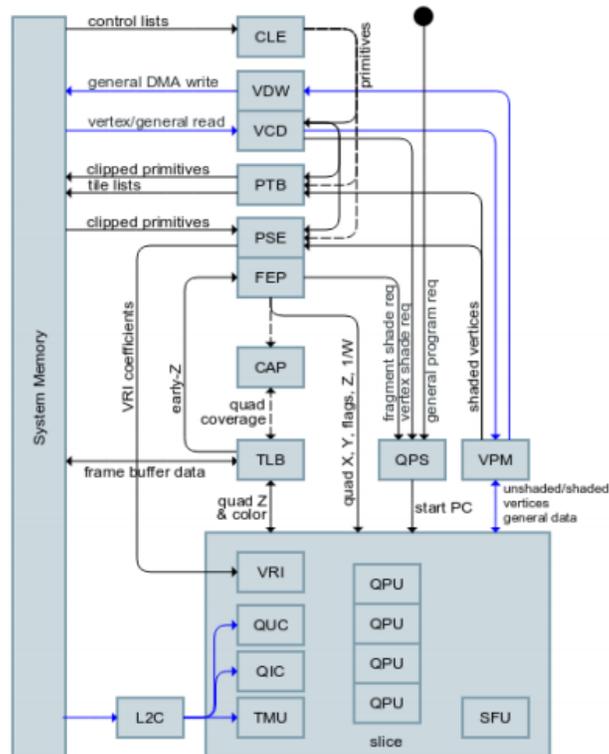


Figura 1.8: Diagrama de Bloques del GPU

Un Slice es un grupo de hasta cuatro QPUs los cuales comparten recursos en común, nombrados Special Function Unit (SFU), Texture Memory Unit (TMU), Varying Interpolation Unit (VRI) y las caches de instrucciones. Los recursos en común son para apoyar QPUs basado en sus funciones de modo que SFU proporciona operaciones matemáticas especiales como recíproco, raíz cuadrada recíproca, logaritmo y exponencial, TMU obtiene información de textura, así como datos de propósito general, VRI variando interpolaciones de coeficientes y caches de instrucciones, conocida como caches Uniforme (QUC) y la memoria cache de instrucciones (QIC), almacena instrucciones de la memoria del sistema.

Cada QPU brinda dos Unidades Aritmético Lógicas (ALUs) para operaciones de suma y multiplicación las cuales son capaces de trabajar en paralelo y soportan datos de tipo entero y punto flotante.

La RAM que podemos encontrar con el SoC depende del modelo. La primera generación ofrece dos tipos de memoria RAM, 256 MB para los modelos A y A+ y 512 MB para los modelos B y B+, en la Raspberry Pi 2 el modelo B, la memoria se ha actualizado a 1 GB y el chip de memoria se trasladó al otro lado de la tarjeta en lugar de apilar en la parte superior del SoC como en los modelos anteriores. La RAM es compartida entre la CPU y la GPU, donde los usuarios pueden configurar la cantidad de RAM proporcionada a la GPU en el software, el resto se le dará a la CPU. El valor mínimo que se debe establecer es de 16 MB y el valor predeterminado es de 64 MB. Es aconsejable dividir la mitad de la memoria de la GPU si la tarjeta funciona en gran medida para tareas multimedia como representación 3D o si se va a conectar con un módulo de cámara.

1.1.3 Paquete de Soporte de Matlab y Simulink para Raspberry Pi

Matlab, integrado con Simulink, es un software de programación desarrollado por MathWorks. Proporciona herramientas de cálculo numérico y modelos utilizados por los académicos e ingenieros para el diseño de varios sistemas y desarrollo. También es ampliamente utilizado en la educación universitaria.

El paquete de soporte de Matlab y Simulink está compuesto de dos partes, una para Matlab y otra para Simulink. Matlab permite el desarrollo de software para los algoritmos que se pueden ejecutar en Raspberry Pi. También permite controlar los dispositivos periféricos conectados a la placa a través de sus interfaces GPIO, es decir, comunicación serie, I2C, SPI, etc. Así como un módulo de cámara a través de funciones en la ventana de comandos del Matlab. El paquete de software para Simulink permite a los usuarios crear modelos de Simulink en diversos campos tales como audio, imagen o sistemas embebidos utilizando toolboxes. Además, los usuarios pueden crear aplicaciones independientes mediante la implementación de

modelos de Simulink en Raspberry Pi (Para mayor información respecto a los comandos de MATLAB® ver Anexo B).

Para que la Raspberry Pi trabaje con MATLAB® y Simulink requiere de la instalación de un Toolbox en el software y un sistema operativo Raspian modificado para tener compatibilidad con la tarjeta.

Para habilitar el Toolbox de MATLAB® y Simulink con todas sus funciones y comandos debemos ejecutar el software como administrador, lo cual nos permitirá control sobre todas las operaciones. Los pasos a seguir para la correspondiente instalación son explicados en el Anexo A.

1.2 Introducción al Procesamiento Digital de Imágenes (PDI)

1.2.1 Definición

El procesamiento digital de imágenes (PDI) se refiere a procesar las imágenes del mundo real de manera digital por medio de un computador. Este es un tema bastante amplio, en el cual se ven involucrados temas de física, matemáticas, ingeniería electrónica, en computación, etc. Esta disciplina estudia los fundamentos conceptuales de la adquisición y algorítmicos del procesamiento como tales. Tiene además, como objetivo mejorar el aspecto de las imágenes y hacer evidentes en ellas ciertos detalles que se desean hacer notas.

La visión artificial (o visión computacional) puede ser definida como el proceso, obtención, caracterización e interpretación de información de imágenes tomadas de un mundo tridimensional. Estos procesos pueden ser subdivididos en seis áreas principales y están agrupados de acuerdo a la complicación y delicadeza que lleva su implementación. Consideramos tres niveles de procesamiento: visión de bajo, medio y alto nivel, como se muestra en la tabla 1.2.

Tabla 1.2: Niveles de Visión y procesos del PDI

Procesos del PDI	Nivel de Visión
1. Captura/adquisición	Bajo
2. Reprocesamiento	
3. Segmentación	Medio
4. Descripción	
5. Reconocimiento	
6. Interpretación	Alto

- **La captura o adquisición:** es el proceso a través del cual se obtiene una imagen digital utilizando un dispositivo de captura como una cámara digital, video cámara, escáner, satélite, etc.
- **El preprocesamiento:** incluye técnicas tales como la reducción del ruido, realce del contraste, realce de ciertos detalles, o características de la imagen.
- **La segmentación:** es el proceso que divide una imagen en objetos que sean de nuestro interés de estudio.
- **La descripción:** es el proceso que obtiene características convenientes para diferenciar un tipo de objeto de otro, como: la forma, el tamaño, área, etc.
- **El reconocimiento:** es el proceso que identifica los objetos, como por ejemplo: una llave, un tornillo, moneda, coche, etc.
- **La interpretación:** es el proceso que asocia un significado a un conjunto de objetos reconocidos (llaves, tornillos, herramientas, etc.) y trata de emular la cognición.

1.2.2 Áreas de aplicación

Las técnicas de procesado de imágenes empezaron a usarse en gran variedad de ámbitos y problemas que compartían la necesidad de mejora de las imágenes para su interpretación y análisis.

En la actualidad existe una inmensa gama de áreas donde el PDI es utilizado de manera habitual. La fuente de imágenes es uno de los criterios de clasificación habitual. La fuente principal de energía de las imágenes es el espectro electromagnético (EM), principalmente la banda de rayos X y el espectro visible. El espectro acústico, ultrasónico y el electrónico son otras fuentes.

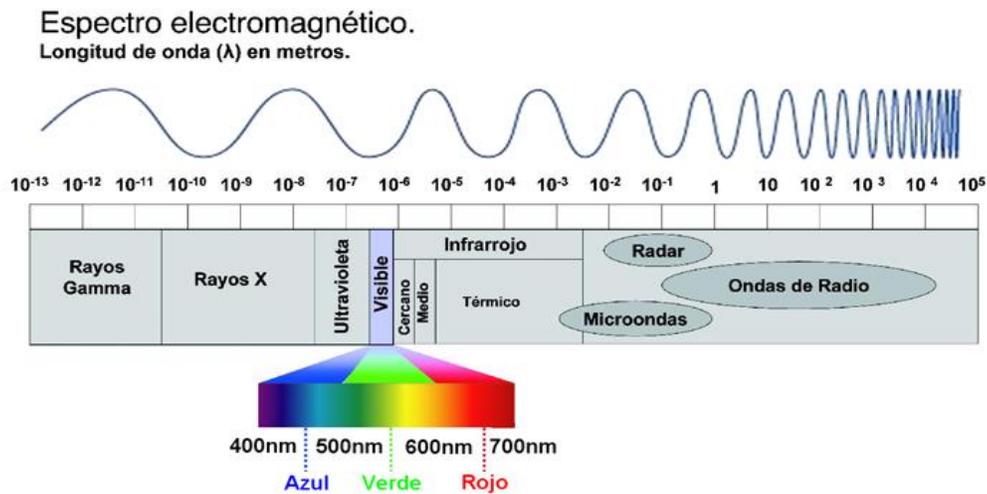


Figura 1.9: Espectro Radioeléctrico (Pregunton, 2014)

Podemos empezar por los rayos gamma, ya que existen aplicaciones para procesar imágenes que vienen de esas fuentes de imagen en la medicina nuclear, como por ejemplo la Tomografía por Emisión de Positrones (PET) donde se inyecta un isótopo radioactivo al paciente que emite rayos gamma que posteriormente se capturan, también tenemos las observaciones astronómicas donde se detecta la radiación natural de rayos gamma de los astros. En la siguiente banda, la de los rayos X, tiene como principales aplicaciones en medicina (radiología, angiografía, Tomografía Axial Computarizada (TAC), etc., pero también en astronomía e industria, como en el control de calidad de placas circuito impreso o de alimentos.

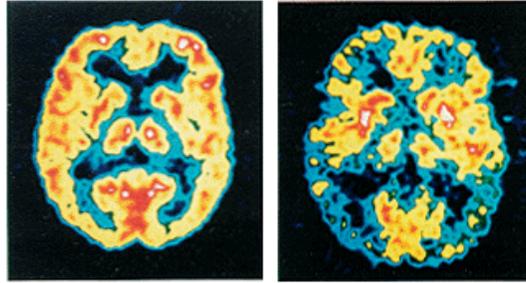


Figura 1.10: Escáner PET (Derecha un cerebro normal, izquierda un cerebro con Alzheimer (Kindsein, 2007)

También tenemos la posibilidad de obtener imágenes empleando la banda ultravioleta del espectro electromagnético, la cual tiene aplicaciones en la litografía, la inspección industrial, la microscopía, los láseres, las imágenes biológicas y la astronomía.

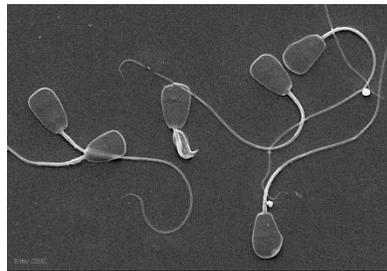


Figura 1.11: Espermatozoides de gacela Mohor observados con microscopía electrónica de barrido.

La banda infrarroja es utilizada normalmente junto con la banda de espectro visible, como por ejemplo en microscopía, astronomía, detección remota (utilizando varias bandas que detectan diferentes detalles del terreno), controles de calidad de la industria y en las aplicaciones de identificación de la policía.



Figura 1.12: Detección de Vegetación con imagen Infrarroja (Sean, 2015)

La banda de microondas tienen como principal aplicación el radar debido a que este puede obtener datos de virtualmente cualquier región y en cualquier momento, sin importar las condiciones climáticas o luz ambiental (puede penetrar nubes, vegetación, hielo, arena) utilizando pulsos de microondas para obtener imágenes a partir de energía reflejada. Las imágenes de la banda de radio tienen su mayor aplicación en astronomía y medicina. En medicina las ondas de radio se usan principalmente en Imágenes de Resonancia Magnética (MIR).



*Figura 1.13: Resonancia Magnética
(Nazareth, 2013)*

Las imágenes electrónicas, como las de microscopía electrónica, la microscopía de escaneo de electrones (SEM) y la microscopía de transmisión de electrones (TEM), utiliza rayos de electrones enfocados en vez de rayos de luz como los microscopios ópticos. Utilizando esta técnica podemos captar un filamento de tungsteno o detectar un circuito integrado dañado.

1.2.3 Conceptos básicos de las imágenes en MATLAB®

MATLAB® es un software matemático el cual nos ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (Lenguaje M). Disponible para todas las plataformas Unix, Windows y Mac OS. Entre sus principales prestaciones encontramos la manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaces de usuario (GUI) y la comunicación (Mathwork, 2015).

MATLAB nos proporciona un entorno de trabajo interactivo cuyo elemento básico de trabajo son las matrices, esto nos permite la resolución de problemas en un tiempo menos que si utilizáramos lenguajes de programación tradicionales, por ejemplo Fortran, Basic o C, con la ventaja que su lenguaje propio de programación es similar a los lenguajes tradicionales. Cuando trabajamos con matrices podemos describir infinidad de variables de manera altamente flexible y matemáticamente eficiente. Por ejemplo, una imagen se puede describir como un matriz de pixeles, un sonido como una matriz de fluctuaciones.

Tabla 1.2: Raspberry Pi, Modelos y Especificaciones

Formato	Descripción	Extensión
BMP	Windows Bitmap	.bmp
GIF	Graphics Interchange Format	.gif
JPEG	Joint Photographic Experts	.jpeg
JPG	Group	
PNG	Portable Network Graphics	.png
TIFF	Tagged Image File Format	.tiff
TIF		
ICO	Windows Icon resources	.ico

En MATLAB® una imagen a escala de grises es representada por medio de una matriz bidimensional de $m \times n$ elementos en donde n representa el número de pixeles de ancho y m el número de pixeles de largo. El elemento v_{11} corresponde al elemento de la esquina superior izquierda, donde cada elemento de la matriz de la imagen tiene un valor de 0 (negro) a 255 (blanco).

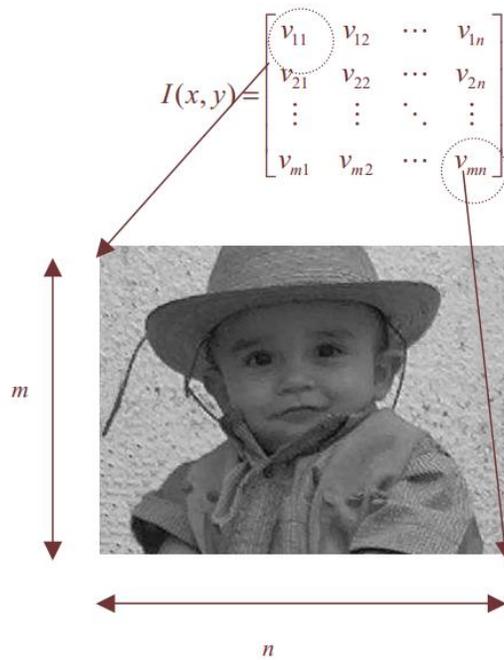


Figura 1.14: Representación de una imagen a escala de grises en MATLAB®

Por otro lado una imagen de color RGB (la más usada para la visión computacional, además de ser para MATLAB® la opción default) es representada por una matriz tridimensional $m \times n \times p$, donde m y n tienen la misma significación que para el caso de las imágenes de escala de grises mientras p representa el plano, que para RGB que puede ser 1 para el rojo, 2 para el verde y 3 para el azul.

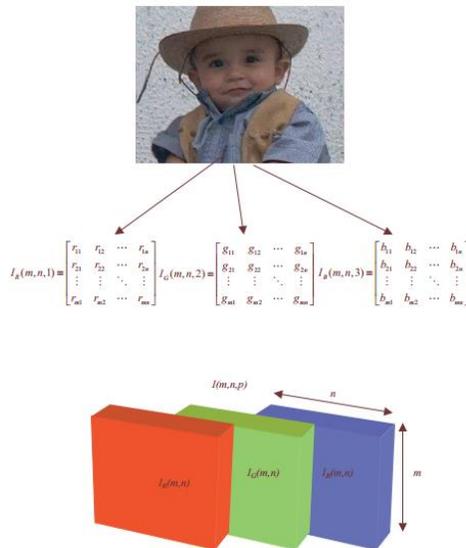


Figura 1.14: Representación de una imagen a color RGB en

1.2.4 Detección de Bordos (EDGE)

Los bordes de una imagen se pueden definir como transiciones entre dos regiones de niveles de gris significativamente distintos. Suministran una valiosa información sobre las fronteras de los objetos y puede ser utilizada para segmentar la imagen, reconocer objetos, etc.

La mayoría de las técnicas para detectar bordes emplean operadores locales basados en distintas aproximaciones discretas de la primera y segunda derivada de los niveles de grises de la imagen.

1.2.4.1 Operadores basados en la primera derivada (Gradiente)

La derivada de una señal continua proporciona las variaciones locales con respecto a la variable, de forma que el valor de la derivada es mayor cuanto más rápidas son estas variaciones.

En el caso de funciones bidimensionales $f(x,y)$, la derivada es un vector que apunta en la dirección de la máxima variación de $f(x,y)$ y cuyo modulo es proporcional a dicha variación. Este vector se denomina gradiente y se define:

$$\nabla f(x, y) = \begin{bmatrix} \frac{\partial f(x, y)}{\partial x} \\ \frac{\partial f(x, y)}{\partial y} \end{bmatrix} \quad (1.1)$$

$$Mag[\nabla f(x, y)] = \sqrt{\left(\frac{\partial f(x, y)}{\partial x}\right)^2 + \left(\frac{\partial f(x, y)}{\partial y}\right)^2} \quad (1.2)$$

$$\theta = \text{arctag} \frac{\frac{\partial f(x, y)}{\partial x}}{\frac{\partial f(x, y)}{\partial y}} \quad (1.3)$$

En el caso bidimensional discreto, las distintas aproximaciones del operador gradiente se basan en diferencias entre los niveles de grises de la imagen. La derivada parcial $f_x(x,y)$ (gradiente de fila) $G_F(i,j)$ puede aproximarse por la diferencia de pixeles adyacentes de la misma fila.

$$\frac{\partial f(x,y)}{\partial x} \approx \nabla f(x,y) = f(x,y) - f(x-1,y)$$

-1	1
----	---

(1.4)

La discretización del vector gradiente en el eje Y ($G_C(i,j)$), será:

$$\frac{\partial f(x,y)}{\partial y} \approx \nabla f(x,y) = f(x,y) - f(x,y-1)$$

-1
1

(1.5)

El gradiente de la fila G_F y de columna G_C en cada punto se obtienen mediante la convolucion de la imagen con las máscaras H_F y H_C , esto es:

$$G_F(i,j) = F(i,j) \otimes H_F(i,j) \tag{1.6}$$

$$G_C(i,j) = F(i,j) \otimes H_C(i,j) \tag{1.7}$$

La magnitud y orientación del vector gradiente suele aproximarse por la expresión:

$$|G(i,j)| = \sqrt{G_F^2 + G_C^2} \approx |G_F(i,j)| + |G_C(i,j)| \tag{1.8}$$

Los operadores más utilizados son los de Roberts, Prewitt y Sobel y Frei-chen.

1.2.4.2 Operador de Roberts

Las máscaras utilizadas en este operador son:

Gradiente fila		
0	0	0
0	0	1
0	-1	0

Gradiente columna		
0	0	0
0	0	1
0	-1	0

Obtiene una respuesta ante bordes diagonales. Ofrece buenas prestaciones en cuanto a localización. El gran inconveniente de este operador es su extrema sensibilidad al ruido y por tanto tiene pobres cualidades de detección.

1.2.4.3 Operadores de Prewitt, Sobel y Frei-Chen

Los tres operadores pueden formularse de forma conjunta con las siguientes mascatas de convolucion a continuación.

$\frac{1}{2+K}$	Gradiente fila	<table style="border-collapse: collapse; text-align: center;"> <tr><td>1</td><td>0</td><td>-1</td></tr> <tr><td>K</td><td>0</td><td>-K</td></tr> <tr><td>1</td><td>-1</td><td>-1</td></tr> </table>	1	0	-1	K	0	-K	1	-1	-1	$\frac{1}{2+K}$	Gradiente columna	<table style="border-collapse: collapse; text-align: center;"> <tr><td>-1</td><td>-K</td><td>-1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>K</td><td>1</td></tr> </table>	-1	-K	-1	0	1	0	1	K	1
1	0	-1																					
K	0	-K																					
1	-1	-1																					
-1	-K	-1																					
0	1	0																					
1	K	1																					

En el operador Prewitt (K=1) se involucran a los vecinos de filas/columnas adyacentes para proporcionar mayor inmunidad al ruido.

El operador Sobel (K=2), se supone que es más sensible a los bordes diagonales que el d Prewitt aunque en la práctica hay poca diferencia entre ellos.

Frei-Chen (K= $\sqrt{2}$), el gradiente es el mismo para bordes verticales, horizontales y diagonales.

La IPT (Image Processing Toolbox) de Matlab dispone de la función *edge*, desde la que se puede detectar los bordes de una imagen utilizando diferentes métodos. La función presenta la estructura siguiente:

$$**img_contorno = edge(im_res)**$$

La función recibe como entrada una imagen binaria o en escala de grises "*im_res*" y devuelve una imagen binaria "*img_contorno*" del mismo tamaño que "*im_res*", con 1's donde la función haya encontrado bordes y 0's en el resto. El algoritmo *edge* dispone de varios métodos para detectar los bordes.

A continuación se enumeran algunos de los diferentes métodos que se pueden utilizar en la función `edge`:

- El método Sobel (método por defecto si no se especifica ninguno en la función), detecta los bordes usando la aproximación Sobel. El operador Sobel calcula el gradiente de la intensidad de una imagen en cada punto (píxel). Así, para cada punto, este operador da la magnitud del mayor cambio posible, la dirección de éste y el sentido desde oscuro a claro. El resultado muestra qué tan abruptamente o suavemente cambia una imagen en cada punto analizado y, en consecuencia, que tan probable es que éste represente un borde en la imagen y, también, la orientación a la que tiende ese borde.
- El método Prewitt detecta los bordes usando la aproximación Prewitt. Suele obtener resultados parecidos a los obtenidos con el método Sobel.
- El método del Laplaciano Gaussiano ('log') detecta los bordes buscando cortes con cero después de filtrar 'img_contorno' con un filtro Laplaciano Gaussiano. Obtiene, normalmente, mejores resultados que el método Sobel pero a costa de tener que modificar parámetros por defecto.
- El método Canny, detecta los bordes buscando máximos locales del gradiente de 'img_contorno'. Este método es el más potente de todos y por ello el más utilizado. Es menos propenso a confundirse por el ruido y más propenso a detectar bordes débiles.
- El algoritmo de detección de bordes de Canny utiliza un filtro basado en la primera derivada de una gaussiana. Ya que es susceptible al ruido presente en datos de imagen sin procesar, la imagen original es transformada con un filtro gaussiano.

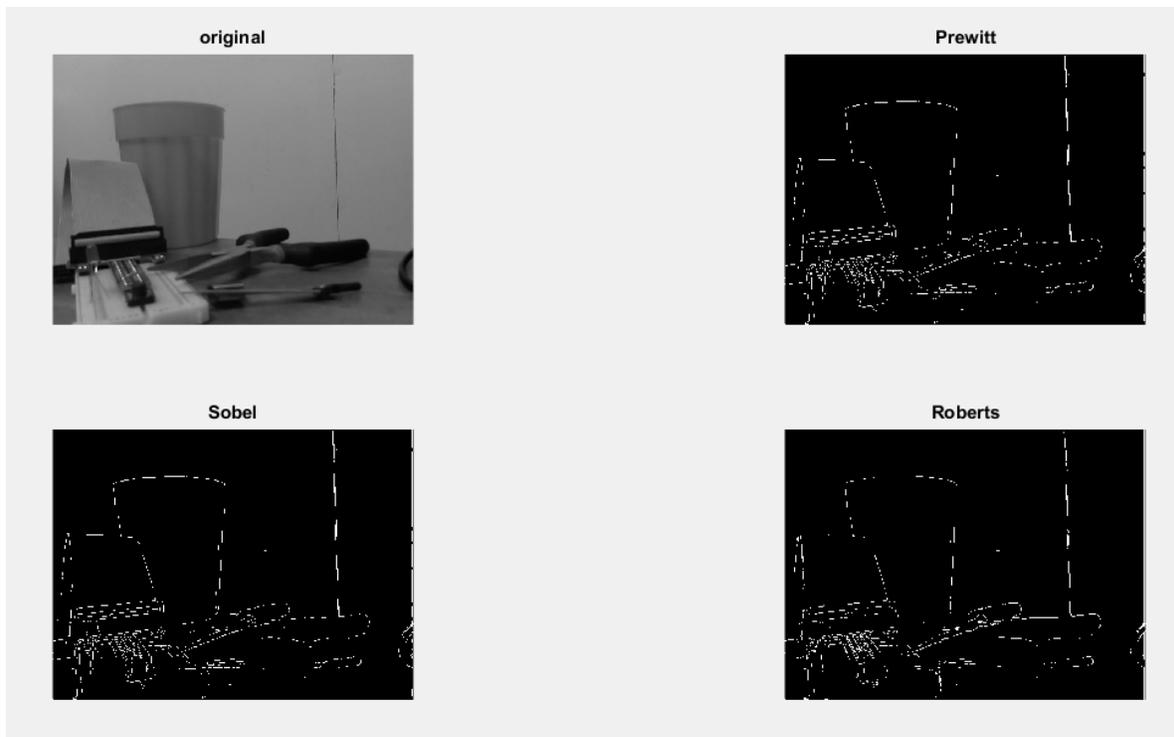


Figura 1.15: Variantes del Filtro Edge

1.2.5 Speeded Up Robust Features (SURF)

El algoritmo SURF (Speeded Up Robust Features), fue planteado por Hebert Bay en el año 2006 (Pina, 2010), se basa en otro algoritmo conocido como SIFT⁴ pero a diferencia de este, es más robusto puesto que analiza wavelets y determinantes Hessianas, además SURF está libre de patentes y existen numerosas implantaciones mucho más eficientes que SIFT.

Hace réplicas de la imagen para así buscar los puntos que estén en todas las réplicas asegurando la invariancia de escala. Las réplicas pueden ser piramidales o del mismo tamaño a la original. En las piramidales la principal idea es tener una imagen más pequeña que la anterior, para ello se realiza un filtro de la imagen y se submuestra ya que si se realiza solo el submuestreo se pierde la información de la imagen como se muestra en la figura 1.16:

(Requena, 2008)⁴ Scale-invariant feature transform



Figura 1.16: Imagen submuestreada

SURF ofrece la detección de objetos con menor coste computacional, es más rápido y eficaz que SIFT en lo que se refiere a realizar cálculos para la generación de puntos clave, todo esto gracias a la reducción de la dimensionalidad y dificultad en el cálculo de vectores (Boullosa, 2011).

Al igual que el método SIFT este utiliza la técnica de Scale-Space $L(x, y, \sigma)$ para reducir el ancho de banda de la imagen original (), a medida que σ va incrementando la imagen se hace más borrosa, como podemos observar en la Figura 1.17:

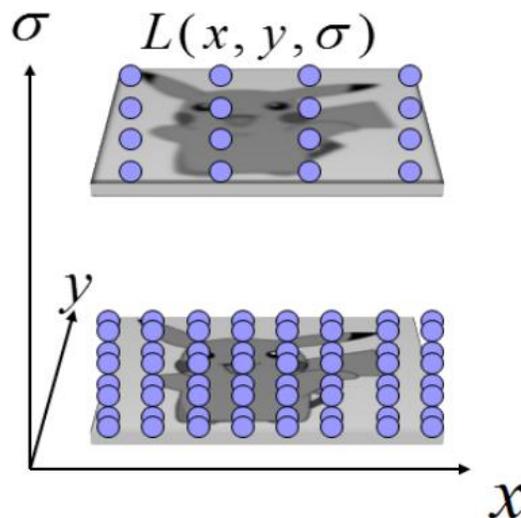


Figura 1.17: Técnica espacio-escala

1.2.5.1 Fundamento Teórico

Como se dijo anteriormente el método SURF está basado en SIFT, siendo este más robusto en cuanto a cálculos. Utiliza determinantes Hessianas para la detección de los puntos de interés localizando a la vez su posición así como su escala, también

utiliza filtros tipo caja para aproximar las derivadas parciales de segundo orden del filtro de Gauss (Ecuación 2.2) y ser evaluados de forma mucho más rápida usando imágenes integrales. El análisis espacio-escala se basa en la elevación del tamaño del filtro (Boullosa, 2011) a diferencia de SIFT que reduce el tamaño del a imagen original para este análisis.

- **Detección de puntos de interés**

Surf utiliza la determinante Hessiana para la localización de la posición y escala de los puntos de interés. La matriz Hessiana $H(x, \sigma)$ permite mayor precisión y rapidez en los cálculos. (Ver ecuación 1.9)

$$H(x, \sigma) = \begin{bmatrix} L_{XX}(x, \sigma) & L_{XY}(x, \sigma) \\ L_{XY}(x, \sigma) & L_{YY}(x, \sigma) \end{bmatrix} \quad (1.9)$$

Teniendo en cuenta que la derivada de segundo orden del filtro de Gauss es:

$$\frac{\partial g(\sigma)^2}{\partial x^2} \quad (2.0)$$

Se realiza una convolucion con la imagen I en el punto $x = (x, y)$ para obtener el valor de $L_{XX}(x, \sigma)$ (Ver ecuaciones 2.3 y 2.4), de igual manera para obtener $L_{XY}(x, \sigma)$ y $L_{YY}(x, \sigma)$ el proceso es similar, estas derivadas son conocidas como derivadas Laplacianas de Gaussianas.

$$\begin{aligned} L_{XX}(x, y, \sigma) &= \frac{\partial g^2}{\partial x^2} * (G(x, y, \sigma) * I(x, y)) \\ &= \left(\frac{\partial g^2}{\partial x^2} * (G(x, y, \sigma))\right) * I(x, y) \end{aligned} \quad (2.1)$$

$$L_{XX}(x, y, \sigma) = G_{XX}(x, y, \sigma) * I(x, y) \quad (2.2)$$

El cálculo de los determinantes de la matriz Hessiana que permite localizar la escala del punto, se define a continuación:

$$\det(H_{\text{aprox}}) = D_{xx}D_{yy} - (\omega D_{xy})^2$$

donde D_{xx} , D_{yy} , D_{xy} son las aproximaciones de las derivadas parciales, ω es el peso para balancear la determinante Hessiana, el cálculo para D_{xx} se realiza a partir de $I(x, y)$ y $G_{xx}(x, y)$, de manera similar D_{yy} , D_{xy} .

El cálculo del peso ω se presenta en la ecuación 2.3, tomando en cuenta la figura 1.17 que realiza filtros tipo caja de 9×9 con aproximaciones Gaussianas de segundo de segundo orden con $\sigma = 1,2$, sugerido en “SURF: Speeded Up Robust Features” (Herbert Bay & Tuytelaars), donde $|X|_F$ es la Norma de Frobenius².

$$\omega = \frac{|L_{XY}(1, 2)|_F |D_{YY}(9)|_F}{|L_{YY}(1, 2)|_F |D_{XY}(9)|_F} = 0,912.. \cong 0,9 \quad (2.3)$$

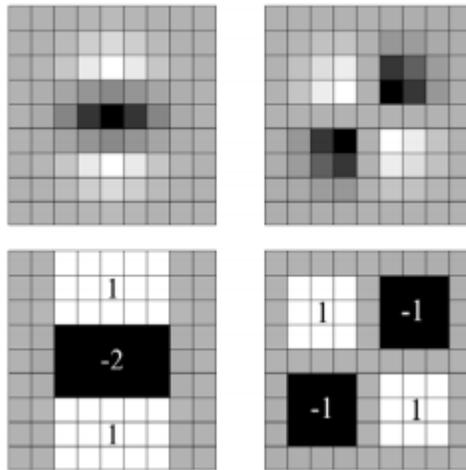


Figura 1.18: Aproximaciones de las derivadas parciales usando filtros tipo caja

La ventaja de SURF es que utiliza cuadrados (Figura 1.18) para promediar la zona de la imagen, utilizando para ello la Imagen Integral (Figura 1.19) descrita en la siguiente ecuación

$$S(x,y) = \sum_{x' \leq x} \sum_{y' \leq y} I(x', y') \quad (2.4)$$

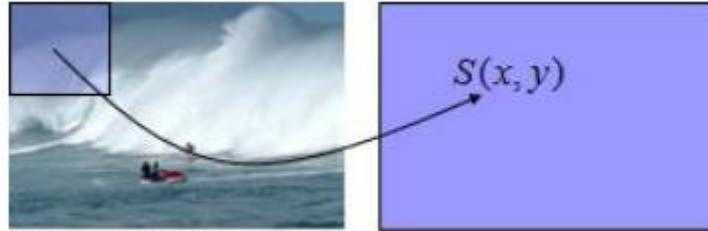


Figura 1.19: Imagen Integral, utilizando cuadrados para promediar la zona de la imagen

Siendo $s(x,y)$ la imagen integral y $I(x', y')$ la imagen original, luego se procede a realizar una convolucion entre la imagen integral y el filtro Haar para encontrar la suma ($\text{Sum} = (1+4) - (2+3)$) del área como se muestra en la Figura 1.20

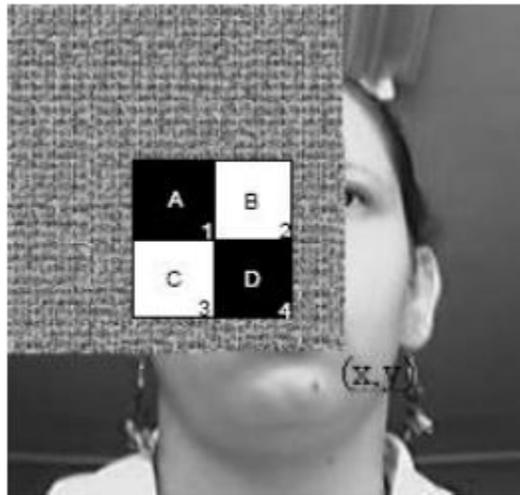


Figura 1.20: Convolucion del filtro Haar con una imagen integral dividida en 4 sub-regiones, donde 1, 2, 3 y 4 son los resultados de la imagen integral.

- **Asignación de Orientación**

Una vez realizada la convolucion y seleccionados los puntos máximos, se debe disponer de la orientación y la descripción de la vecindad. Se establece un descriptor para cada punto con la ayuda de los filtros Haar que nos brindan una solución en los ejes x y y en una región circular de $6s$ usado por el autor (Herbert Bay & Tuytelaars) , siendo s la escala del punto de interés.

Con los cálculos realizados para todo los vecinos se estima la orientación dominante, se procede a sumar todos los resultados dentro de una ventana deslizante que cubre un ángulo de $\pi/3$ recomendado en (Herbert Bay & Tuytelaars), ver Figura 1.21

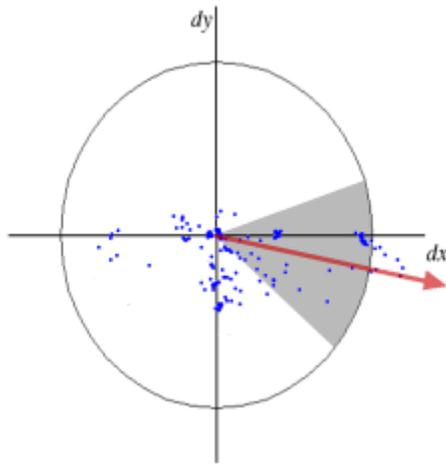


Figura 1.21: Asignación de orientación

- **Creación del descriptor**

Se crea una región cuadrada centrada en el punto de interés de tamaño 20s como se observa en la Figura 1.22. La región es subdividida en subregiones de 4x4, en cada una de las cuales se calculan las respuestas de Haar siendo estas d_x y d_y , para cada sub-región se suman los resultados así como sus valores absolutos $|d_x|$, $|d_y|$ como se muestra en la figura 19 quedando formado un vector para cada una de



Figura 1.22: Imagen con las regiones cuadradas centradas alrededor de los puntos de interés de tamaño 20s

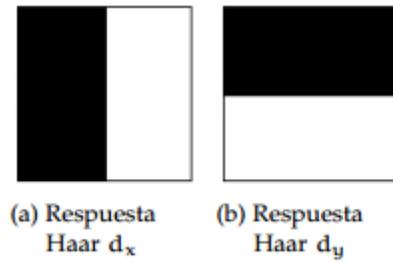


Figura 1.23: Respuesta Haar de tamaño de $2s$, la parte de color negro representa un peso de -1 y el blanco 1

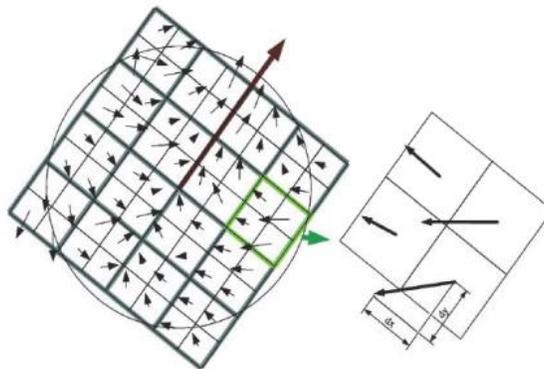


Figura 1.24: Respuesta de Haar en cada una de las Sub-regiones alrededor del punto de interés

Las regiones presentadas en la ecuación 2.5, los puntos resultantes son invariantes a la escala.

$$v = \left(\sum dx, \sum dy, \sum |dx|, \sum |dy| \right) \quad (2.5)$$

De esta forma se obtiene un descriptor de 64 valores para cada punto identificado.

En la figura 1.25 se presenta una imagen con los resultados obtenidos después de aplicar el método SURF:

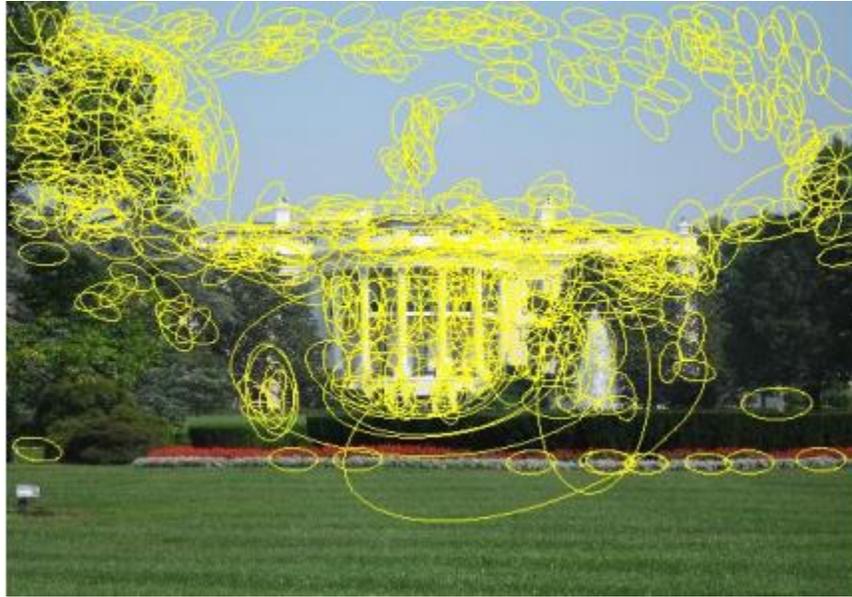


Figura 1.26: Resultado aplicando el metodo SURF

1.2.6 Histogramas de Gradientes Orientados (HOG) y descriptores de HOG

El Histograma de Gradientes Orientados de una imagen (del inglés *Histogram of Oriented Gradients* - HOG) tiene como rango de valores posibles las distintas orientaciones que pueden tomar los gradientes de los píxeles, i.e. los distintos grados que pueden tomar sus ángulos de gradiente (e.g. $[-90^\circ, 90^\circ]$, $[0^\circ, 180^\circ]$, $[0^\circ, 360^\circ]$...). Este rango se divide en sub-clases – del mismo tamaño o distintos – (e.g. para el rango $[0^\circ, 180^\circ]$, dividiendo en nueve sub-rangos: $[0^\circ, 20^\circ)$, $[20^\circ, 40^\circ)$... $[160^\circ, 180^\circ]$), y se almacena en cada uno de ellas la suma de las magnitudes de gradiente de los píxeles cuyo ángulo de gradiente se encuentra comprendido entre esos valores.

Partiendo del concepto de los HOG, se puede obtener más información de una imagen por medio de un Descriptor de HOG (del inglés HOG Descriptor). En éste, la imagen se divide en un cierto número de sub-imágenes del mismo tamaño, denominadas *celdas*, y éstas se agrupan en *bloques* con un mismo número de celdas de ancho y alto todos ellos. Además, estos bloques se encuentran solapados de forma que el avance de bloques horizontalmente se realiza eliminando la columna de celdas de la izquierda y añadiendo la columna de la derecha y,

verticalmente, eliminando la fila de celdas de arriba y añadiendo la fila de celdas de abajo. De este modo, dados una imagen de tamaño; un tamaño de celda con w ; y un tamaño de bloque en celdas; el ancho y alto de la imagen en celdas, w_c y h_c , y el número de bloques distribuidos horizontalmente y verticalmente, n_{BW} y n_{BH} , se calculan:

$$w_c = \frac{W}{C_w}, h_c = \frac{H}{C_h} \quad (3.1)$$

$$n_{BW} = 1 + w_c - B_w, n_{BH} = 1 + h_c - B_h \quad (3.2)$$

Y por tanto, el número total de celdas N_c y el número total de bloques N_B resultantes de la imagen A será igual a:

$$N_c = w_c * h_c \quad (3.3)$$

$$N_B = n_{BW} * n_{BH} \quad (3.4)$$

Y la distribución de celdas y bloques es la siguiente:

$$A = \begin{pmatrix} a_{00} & a_{10} & \dots & a_{(w-1)0} \\ a_{01} & a_{11} & \dots & a_{(w-1)1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{0(h-1)} & a_{1(h-1)} & \dots & a_{(w-1)(h-1)} \end{pmatrix} \quad (3.5)$$

$$C = \begin{pmatrix} C_{00} & C_{10} & \dots & C_{(w_c-1)0} \\ C_{01} & C_{11} & \dots & C_{(w_c-1)1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{0(h_c-1)} & a_{1(h_c-1)} & \dots & a_{(w_c-1)(h_c-1)} \end{pmatrix} \quad (3.6)$$

$$B = \begin{pmatrix} B_{00} & B_{10} & \dots & B_{(n_{BW}-1)0} \\ B_{01} & B_{11} & \dots & B_{(n_{BW}-1)1} \\ \vdots & \vdots & \ddots & \vdots \\ B_{0(n_{BH}-1)} & B_{1(n_{BH}-1)} & \dots & B_{(n_{BW}-1)(n_{BH}-1)} \end{pmatrix} \quad (3.7)$$

A partir de esta estructuración de la imagen, el descriptor de HOG calcula de forma independiente el HOG de cada celda y cada bloque agrupa los HOGs de sus celdas correspondientes. Entonces, el número de HOGs que contiene un descriptor será:

$$N_{HOG} = B_W * B_H * N_B \quad (3.8)$$

Y si se divide el HOG en “n” clases, dado que cada bloque contiene $B_W * B_H$ HOGs, entonces el número total de valores N_V que se tomara de la imagen A será:

$$N_V = n * N_{HOG} \quad (3.9)$$

Para el cálculo del modelo de detección, sobre la colección de imágenes se debe calcular el descriptor de HOG de cada imagen, etiquetando cada descriptor como positivo si es una imagen de mano (“+1”) o negativo si no lo es (“-1”).

El cálculo de los descriptores de HOG puede tener un coste en tiempo de computación bastante grande pues requiere del cálculo de un HOG por cada celda. Para agilizar esto se puede usar una técnica denominada Integral de HOG por el cual se mejora el coste de computación a cambio de aumentar el coste en memoria. Para calcular la Integral de HOG se necesitan primero matrices auxiliares de tamaño igual a la imagen de entrada A , $W \times H$, y donde n es el número de clases en el HOG. Cada una de estas matrices estará asociada a una única clase, es decir, a un único rango de ángulos de gradiente, y donde cada uno de sus píxeles cumple:

$$M(b, i, j) = \begin{cases} |G(i, j)|, & \theta(i, j) \in R(b) \\ 0, & eoc \end{cases} \quad (3.10)$$

Donde $b \in \{0, 1, \dots, n-1\}$ indica la clase, $i \in \{0, 1, \dots, W-1\}$ indica la columna de la matriz, $j \in \{0, 1, \dots, H-1\}$ indica la fila de la matriz y $R(b)$ es el rango de valores de ángulo asociado a la clase b . En la figura se puede ver una representación gráfica de estas matrices.

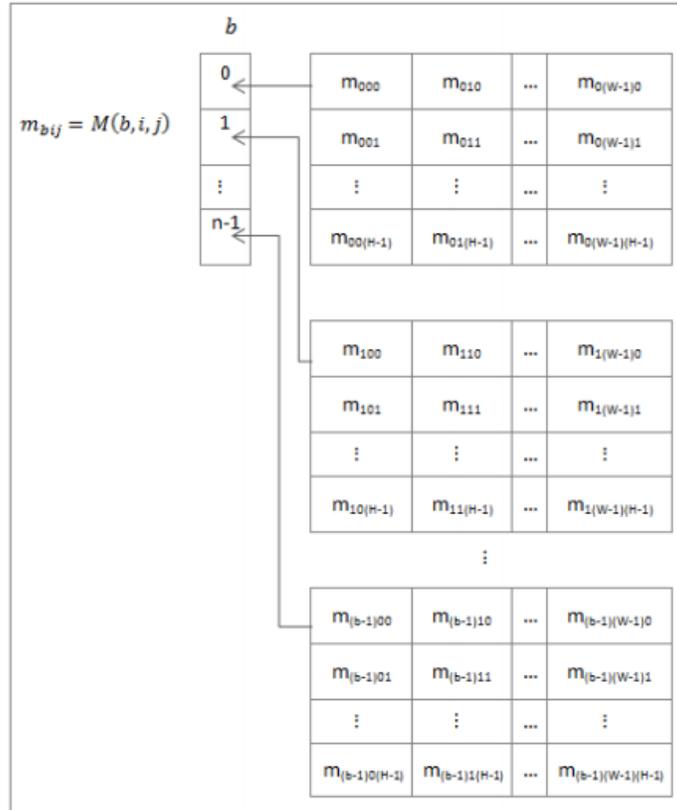


Figura 1.27: Matrices auxiliares para el cálculo de una Integral de HOG

La Integral de HOG (véase ilustración) está compuesta de matrices de tamaño $(W + 1) \times (H + 1)$, cada una de ellas asociadas a una clase y donde cada píxel (i, j) indica la suma de las magnitudes del rectángulo de la imagen A con esquina superior izquierda $(0, 0)$ y tamaño $i \times j$. Estas matrices se calculan a partir de las matrices anteriores y se puede definir una Integral de HOG como la siguiente función recursiva:

$$IH(b, i, j) = \begin{cases} 0, & i = 0 \vee j = 0 \\ IH(b, i, j - 1) + \sum_{k=0}^i B(b, k, j), & eoc \end{cases} \quad (3.11)$$

De esta forma, para el cálculo del HOG de cada celda de un Descriptor de HOG se deberá realizar una simple resta por cada clase por medio de la siguiente ecuación:

$$C(b, i, j, Wc, Hc) = IH(b, i + Wc, j + Hc) - IH(b, i + Wc, j) - IH(b, i, j + Hc) + IH(b, i, j) \quad (3.12)$$

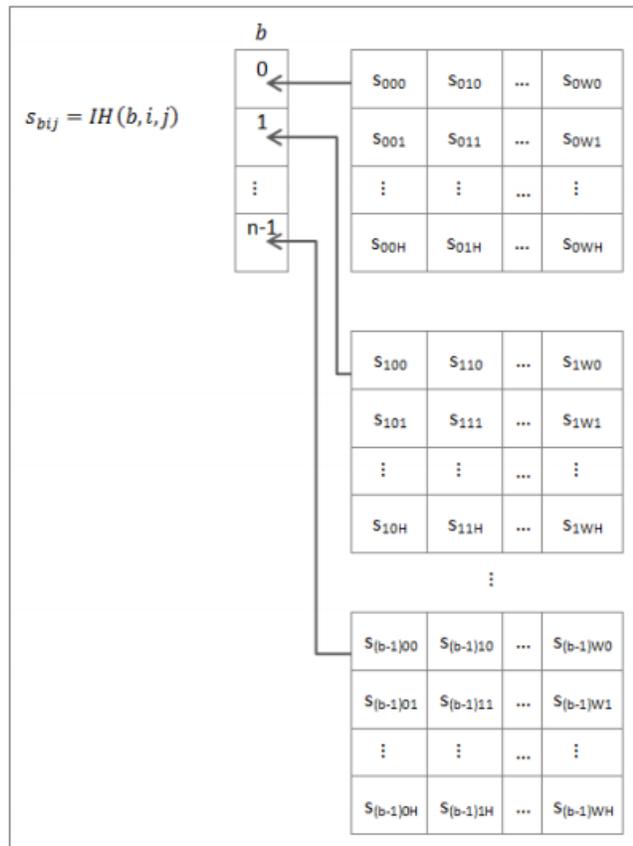


Figura 1.28: Integral de HOG

1.2.7 Máquina de vectores de soporte (SVM)

Una máquina de vectores de soporte (del inglés Support Vector Machine - SVM) es un método de aprendizaje supervisado que, a partir de unas muestras de entrada, reconoce patrones y permite resolver problemas de clasificación y de regresión. En este caso, se usará este algoritmo para poder identificar manos, el cual se trata de un problema de clasificación de dos clases: una clase asociada a las manos y otra a todo aquello que no lo sea.

Se puede concebir de una forma abstracta el problema de clasificación de dos clases como se aprecia en la ilustración. Partiendo de un conjunto de muestras iniciales positivas y negativas con características cada uno, se posiciona cada muestra como un punto en un espacio de dimensiones. El objetivo es encontrar, si

es posible, una línea o hiperplano que separe los puntos de ambas clases en dos grupos, y haciendo que la distancia global entre los puntos de las mismas sea lo máximo posible.

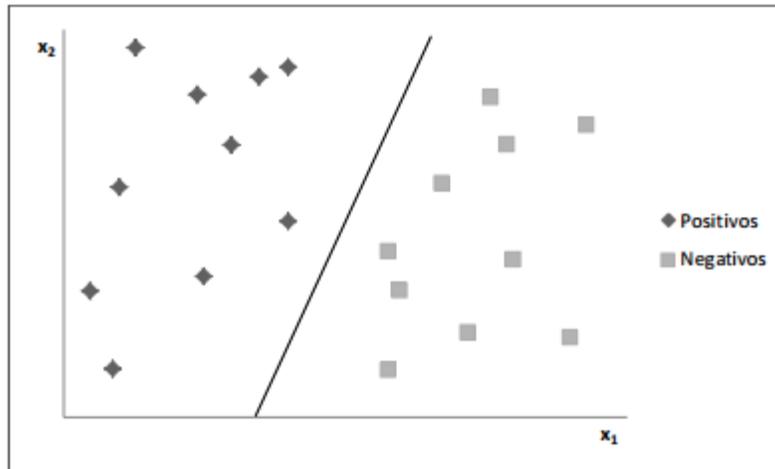


Figura 1.29: Hiperplano de separación de dos clases

Ahora se procederá a profundizar un poco más en las SVM realizando una descripción matemática del funcionamiento de una máquina de soporte para este caso particular de clasificación en dos clases.

Dada una entrada compuesta por M muestras de entrenamiento $X_i \in \mathbb{R}^n$ ($i = 1, \dots, M$) pertenecientes a las clases C_1 o C_2 y las etiquetas asociadas $y_i \in \{1, -1\}$, siendo $y_i = 1$ si x_i pertenece a la clase C_1 y $y_i = -1$ si pertenece a la clase C_2 , se dice que las M muestras son *linealmente separables* si podemos determinar su *función de decisión* como:

$$D(x) = w \cdot x + b \quad (4.1)$$

Donde $w \in \mathbb{R}^n$, $b \in \mathbb{R}$ y se cumple que para $i = 1, \dots, M$

$$w \cdot x_i + b \begin{cases} > 0 & \text{para } y_i = 1 \\ < 0 & \text{para } y_i = -1 \end{cases} \quad (4.2)$$

Suponiendo que los datos de entrada son linealmente separables, ninguna entrada X_i satisface $w \cdot X_i + b = 0$. Para controlar la separabilidad entre las clases y poder

separar las muestras en función del valor de su etiqueta, se sustituyen las anteriores desigualdades por las siguientes:

$$w \cdot x_i + b \begin{cases} \geq 1 & \text{para } y_i = 1 \\ \leq -1 & \text{para } y_i = -1 \end{cases} \quad (4.3)$$

Simplificando las ecuaciones de desigualdad de (4.3):

$$y_i(w \cdot x_i + b) \geq 1 \quad \text{para } i = 1, \dots, M \quad (4.3)$$

El hiperplano

$$D(x) = w \cdot x + b = c \quad \text{para } -1 < c < 1 \quad (4.3)$$

crea un hiperplano de separación entre los datos de entrada X_i ($i = 1, \dots, M$). Cuando $c = 0$, el hiperplano de separación se encuentra en el medio de los hiperplanos con $C = 1$ y $C = -1$. Se denomina *margen* a la distancia existente entre el hiperplano de separación y la muestra de entrenamiento más cercana al mismo. Suponiendo que los hiperplanos $D(x) = 1$ y $D(x) = -1$ contienen al menos una muestra cada una, $D(x) = 0$ tiene el máximo margen para $-1 < c < 1$. El hiperplano con máximo margen se denomina *óptimo hiperplano de separación*.

El vector es ortogonal al hiperplano de separación y, por tanto, la línea que forman una muestra de entrada y el vector es también ortogonal a la misma. Esta línea viene definida por la siguiente ecuación: $\frac{d}{\|w\|} w + x$ donde $|d|$ es la distancia entre X y el hiperplano. Esta línea cortará el hiperplano de separación cuando se satisfaga

$$D\left(\frac{d}{\|w\|} w + x\right) = 0 \quad (4.4)$$

Despejando esta ecuación se obtiene el valor de la distancia d :

$$\begin{aligned} D\left(\frac{d}{\|w\|} w + x\right) &= w^T \cdot \left(\frac{d}{\|w\|} w + x\right) + b = \frac{d}{\|w\|} w^T \cdot w + w^T \cdot x + b \\ &= \frac{d}{\|w\|} \|w\|^2 + w^T \cdot x + b = d\|w\| + w^T \cdot x + b = d\|w\| + D(x) = 0 \end{aligned}$$

$$\Rightarrow d = -\frac{D(x)}{\|w\|}$$

Por tanto, para los hiperplanos $D(x) = 1$ y $D(x) = -1$, su distancia al hiperplano de separación es $\frac{1}{\|w\|}$ en ambos casos por lo que la distancia entre ambos hiperplanos es $\frac{2}{\|w\|}$.

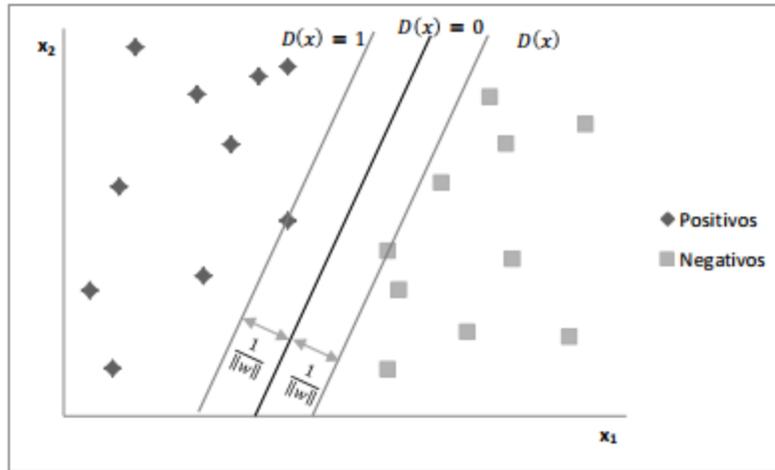


Figura 1.30: Hiperplano de separación óptimo y sus distancias de margen

El objetivo será maximizar esta distancia por lo que se está ante el siguiente problema de optimización:

$$\begin{aligned} \text{maximizar } f(w,b) &= \frac{2}{\|w\|} \\ \text{tal que } y_i(w \cdot x + b) &\geq 1 \quad \text{para } i = 1, \dots, M \end{aligned} \tag{4.5}$$

Un problema de *programación cuadrática* (del inglés quadratic programming-QP) es un tipo especial de problema de optimización matemática en el cual se dispone de una función objetivo de orden cuadrático de múltiples variables que se quiere optimizar (maximizar o minimizar) y que se encuentra sujeto a una serie de restricciones sobre esas variables de orden lineal. La forma general de un QP-problema de minimización es el siguiente:

$$\begin{aligned} \text{minimizar } g(x) &= \frac{1}{2}x^T Qx + c^T x & (4.6) \\ \text{tal que } Ax &\leq b \text{ con } x \geq 0 \end{aligned}$$

Donde x es el vector n -dimensional de las n variables a calcular, c es un vector n -dimensional que tiene por valor los coeficientes de los términos lineales de la función objetivo, Q es una matriz simétrica $n \times n$ que incluye los coeficientes de los términos cuadráticos de la función y las restricciones son definidas por una matriz $n \times m$ A de coeficientes aplicadas al vector de entrada y un vector b m -dimensional de valores para la desigualdad. Se supone que existe una solución que satisface todas las restricciones.

Entonces, se puede convertir el problema de optimización (4.5) en el siguiente QP problema sobre w y b :

$$\begin{aligned} \text{minimizar } g(w, b) &= \frac{1}{2} \|w\|^2 & (4.7) \\ \text{tal que } y(w \cdot x + b) &\geq 1 \quad \text{para } i = 1 \dots M \end{aligned}$$

Dado que se ha supuesto que los datos de entrada son linealmente separables, se sabe que existen w y b que satisfacen (4.8). A las soluciones que satisfacen (4.8) se las denomina *soluciones factibles*. La solución a las restricciones puede no ser única pero el valor de la solución objetivo sí que lo es.

El método de los *multiplicadores de Lagrange* permite convertir un problema de optimización restringido de n variables a otro no restringido de $n + m$ variables donde m es el número de restricciones. Cada una de estas variables asociadas a las restricciones se las denomina *multiplicador de Lagrange*.

Para que el problema sea más fácil de manejar, se convierte la expresión restringida (4.5) y (4.6) por una expresión sin restricción usando multiplicadores de Lagrange:

$$\begin{aligned}
h(w, b, \alpha) &= \frac{1}{2} \|w\|^2 - \sum_{i=1}^M \alpha_i \{y_i(w \cdot x_i + b) - 1\} \\
&= \frac{1}{2} \|w\|^2 - \sum_{i=1}^M \alpha_i y_i (w \cdot x_i + b) + \sum_{i=1}^M \alpha_i
\end{aligned} \tag{4.8}$$

Donde $\alpha = (\alpha_1, \dots, \alpha_M)$ y α_i es un multiplicador de Lagrange no negativo.

La solución óptima de (4.7) se da cuando es minimizada con respecto a W , maximizada con respecto a α_i y maximizada o minimizada con respecto a b dependiendo del signo de $\sum_{i=1}^M \alpha_i y_i$.

El método de Karush-Kuhn-Tucker (KKT) es utilizado para derivar condiciones que comprueban si una solución dada es óptima o no. Aplicándolo a este problema se obtienen las siguientes condiciones de (KKT):

$$\frac{\partial h(w, b, \alpha)}{\partial w} = 0 \tag{4.9}$$

$$\frac{\partial h(w, b, \alpha)}{\partial b} = 0 \tag{4.10}$$

$$\alpha_i \{y_i(w \cdot x_i + b) - 1\} = 0 \quad \text{para } i = 1, \dots, M \tag{4.11}$$

$$\alpha_i \geq 0 \quad \text{para } i = 1, \dots, M \tag{4.12}$$

Por (4.10) y (4.11) se sabe que, o bien $\alpha_i = 0$, o bien $\alpha_i \neq 0$ y $y_i(w \cdot x + b) = 1$. Aquellas muestras de entrada x_i con $\alpha_i \neq 0$ son los denominados *vectores de soporte*.

A partir de (4.7), se deduce con (4.8) y (4.9) respectivamente que

$$\frac{\partial h(w, b, \alpha)}{\partial w} = w - \sum_{i=1}^M \alpha_i y_i x_i = 0 \Rightarrow w = \sum_{i=1}^M \alpha_i y_i x_i \tag{4.13}$$

Y

$$\frac{\partial h(w, b, \alpha)}{\partial b} = \sum_{i=1}^M \alpha_i y_i = 0 \tag{4.14}$$

Sustituyendo (4.12) y (4.13) en (4.7) se obtiene

$$\begin{aligned}
 h(w, b, \alpha) &= \frac{1}{2} \sum_{i=1}^M \alpha_i y_i x_i \sum_{j=1}^M \alpha_j y_j x_j - \sum_{i=1}^M \alpha_i y_i \left(\sum_{j=1}^M \alpha_j y_j x_j \cdot x_i + b \right) \\
 &\quad + \sum_{i=1}^M \alpha_i \\
 &= \frac{1}{2} \sum_{i,j=1}^M \alpha_i \alpha_j y_i y_j x_i x_j - \sum_{i,j=1}^M \alpha_i \alpha_j y_i y_j x_i x_j - \sum_{j=1}^M \alpha_j y_j b \\
 &\quad + \sum_{i=1}^M \alpha_i = -\frac{1}{2} \sum_{i,j=1}^M \alpha_i \alpha_j y_i y_j x_i x_j - 0 + \sum_{i=1}^M \alpha_i \\
 &\Rightarrow h(\alpha) = \sum_{i=1}^M \alpha_i - \frac{1}{2} \sum_{i,j=1}^M \alpha_i \alpha_j y_i y_j x_i x_j
 \end{aligned} \tag{4.15}$$

Se está por tanto ante el siguiente problema de optimización dual:

$$\text{maximizar } h(\alpha) = \sum_{i=1}^M \alpha_i - \frac{1}{2} \sum_{i,j=1}^M \alpha_i \alpha_j y_i y_j x_i x_j \tag{4.16}$$

$$\text{tal que } \sum_{i=1}^M \alpha_i y_i = 0, \quad \alpha_i \geq 0 \quad \text{para } i = 1, \dots, M \tag{4.17}$$

Resolviendo este problema (mediante programación dinámica por ejemplo) se obtiene una solución para los multiplicadores de Lagrange $\alpha = (\alpha_1, \dots, \alpha_M)$ y todas aquellas muestras x_i asociadas a un α_i positivo serán los vectores de soporte. Entonces, a partir de (4.12) la función de decisión viene dada por

$$D(x) = \sum_{i \in S} \alpha_i y_i x_i \cdot x + b \tag{4.18}$$

donde S es el conjunto de índices de las muestras que son vectores de soporte, y de la condición KKT (4.10) se deduce

$$y_i(w \cdot x_i + b) = 1 \quad (4.19)$$

$$b = \frac{1}{y_i} - w \cdot x_i = y_i - w \cdot x_i$$

Y finalmente se tiene ya una función de decisión que para una muestra indefinida x realizará la clasificación del siguiente modo:

$$\begin{cases} \text{Clase } C_1 & \text{si } D(x) > 0 \\ \text{Clase } C_2 & \text{si } D(x) < 0 \end{cases} \quad (4.20)$$

Si se cumple $D(x) = 0$, entonces la muestra x se encuentra en la frontera y es inclasificable.

El caso explicado es el denominado *Máquina de vectores de soporte de margen duro* (del inglés *Hard-Margin*) en el cual los datos de entrada son separados linealmente sin problemas. Sin embargo, cuando no se pueden separar linealmente, este método no da soluciones factibles. Añadiendo una cierta compensación entre la maximización del margen y la minimización del error de clasificación se puede solucionar este problema. Se está entonces ante una *Máquina de vectores de soporte de margen blando* y a esta compensación se la denomina margen C .

Se puede mejorar el tiempo de computación que requieren las *Máquina de vectores de soporte* para su entrenamiento realizando el cálculo con ayuda de kernels. Estos kernels pueden ser de múltiples tipos y cada uno de ellos ofrece resultados distintos: lineales, polinomiales, funciones de base radial, redes neuronales de tres niveles, etc

1.3 Enfoque Constructivista

La teoría constructivista se enfoca en la construcción del conocimiento a través de actividades basadas en experiencias ricas en contexto. El constructivismo nos ofrece un nuevo paradigma para esta nueva era de información motivado por las nuevas tecnologías que han surgido en los últimos años. Actualmente, los esquemas están cambiando, las nuevas tecnologías están causando repercusión en el método de aprendizaje de los estudiantes, lo cual debería provocar transformaciones en la metodología de la enseñanza (Requina, 2008).

1.3.1 Características del aprendizaje constructivista

El ambiente de aprendizaje se diferencia por estas características:

1. El ambiente constructivista en el aprendizaje provee a las personas del contacto con múltiples representaciones de la realidad.
2. Las múltiples representaciones de la realidad evaden las simplificaciones y representan la complejidad del mundo real.
3. El aprendizaje constructivista se enfatiza al construir conocimiento dentro de la reproducción del mismo.
4. El aprendizaje constructivista resalta tareas auténticas de una manera significativa en el contexto en lugar de instrucciones abstractas fuera del contexto.
5. El aprendizaje constructivista proporciona entornos de aprendizaje como entornos de la vida diaria o casos basados en el aprendizaje en lugar de una secuencia predeterminada de instrucciones.
6. Los entornos de aprendizaje constructivista fomentan la reflexión en la experiencia.
7. Los entornos de aprendizaje constructivista permiten el contexto y el contenido dependiente de la construcción del conocimiento
8. Los entornos de aprendizaje constructivista apoyan la «construcción colaborativa del aprendizaje, a través de la negociación social, no de la competición entre los estudiantes para obtener apreciación y conocimiento».

En los últimos diez años, muchos investigadores han explorado el papel que puede desempeñar la tecnología en el aprendizaje constructivista, demostrando que los ordenadores proporcionan un apropiado medio creativo para que los estudiantes se expresen y demuestren que han adquirido nuevos conocimientos. Los proyectos de colaboración en línea y publicaciones web también han demostrado ser una manera nueva y emocionante para que los profesores comprometan a sus estudiantes en el proceso de aprendizaje.

CAPITULO II. ANALISIS Y PRESENTACION DE LOS RESULTADOS

2.1 Diseño Metodológico

El trabajo realizado para esta monografía se realizó en tres etapas. También se contó con un grupo de aproximadamente 10 estudiantes con los cuales se realizó la prueba de algunas de las guías desarrolladas.

En la primera etapa se realizaron experimentos que permitieran al equipo conectarse a una red o directamente a una computadora. Una vez finalizados los experimentos de conexión, se trabajó con el software MATLAB® realizando pruebas procesando imágenes guardadas en la computadora, una vez hecho esto se procedió a usar la Raspberry Pi con el módulo de cámara dedicado que tenemos disponible para ella. Para finalizar esta etapa se realizaron pruebas en tiempo real.

La segunda etapa consistió en el planteamiento y desarrollo de medios de laboratorio, utilizando tres algoritmos distintos de procesamiento de imágenes, siendo estos:

- El algoritmo EDGE
- El algoritmo SURF
- Método de HOG y SVM

Finalizando con el desarrollo de cuatro métricas de laboratorio contando con la participación de diez estudiantes quienes no tenían conocimientos previos en lo que se refiere a esta disciplina.

En la etapa final, se implementaron y evaluó el contenido desarrollado. Debido a la falta de equipos disponibles se realizó con un grupo de pequeño de estudiantes.

Las herramientas utilizadas en esta etapa fueron:

- Encuestas aplicadas a los estudiantes para determinar el nivel de conocimientos que se tenían en esta disciplina y las herramientas a utilizar.
- Entrevistas a docentes de la Facultad de Ingeniería Electrónica.
- Encuesta final aplicada a los estudiantes.
- Software estadístico SPSS

A continuación se explican los resultados obtenidos en cada etapa del proyecto.

2.2 Análisis de la problemática.

2.2.1 Encuestas aplicadas a los estudiantes que participaron como grupo de prueba para las guías desarrolladas.

Este punto consistió en determinar las opiniones generales respecto a los conocimientos que los estudiantes tenían respecto al tema, para esto se consultaron a estudiantes que cursaron la asignatura de Electrónica Aplicada, Sistemas de Control, Sistemas de Medición, Señales y Sistemas, Electrónica Digital 1 y 2 con año de ingreso 2013.

El medio utilizado para que los estudiantes completaran la encuesta fue a través de la página web www.encuestafacil.com, esta nos brinda distintos formatos para encuestas y a la misma vez genera los resultados de las encuestas aplicadas. Se utilizó este medio de recolección de datos debido a que la mayoría de los estudiantes no se pudo tener contacto con los estudiantes personalmente, por lo que se decidió utilizar dicha herramienta.

A través de correos electrónicos y contactos en Facebook se facilitó el enlace de la encuesta obteniendo una respuesta de 10 estudiantes.

El gráfico muestra que la mayoría de los estudiantes 2010 – 2013 carecen de conocimientos en el área de procesamiento digital de imágenes, teoría, técnicas y aplicaciones. En este punto la opción más elegida fue “No” que representa un 70% de los estudiantes.

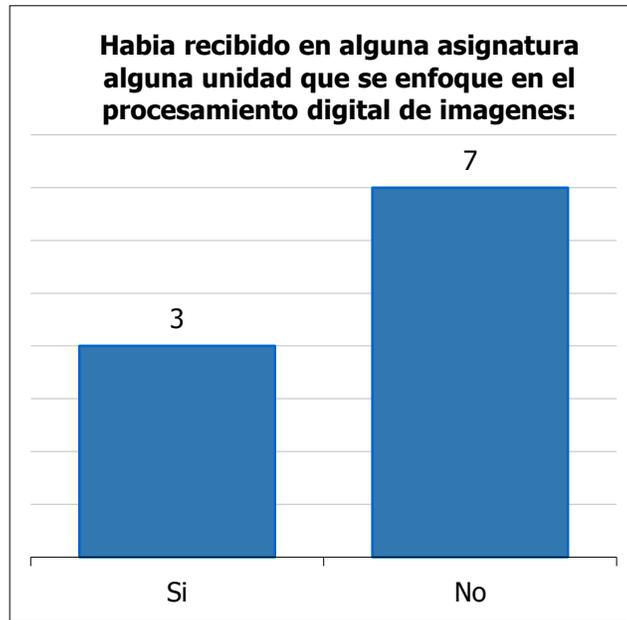
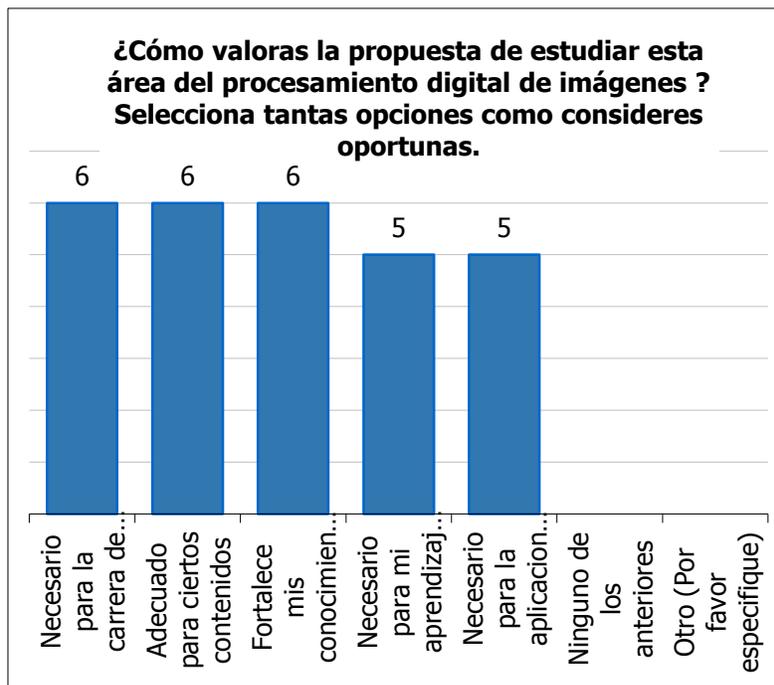


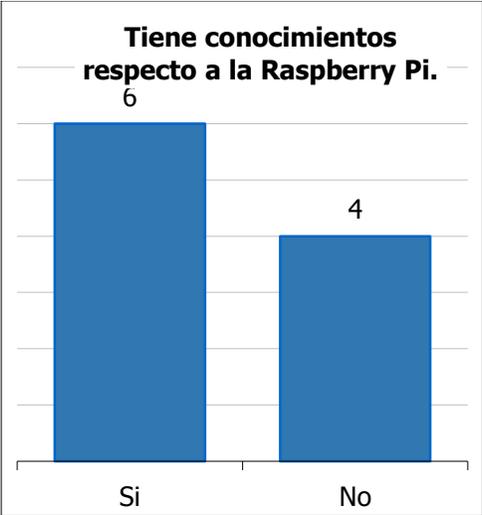
Grafico 1: Resultados de los estudiantes encuestados

El resultado en el gráfico 2 muestra las consideraciones de los estudiantes, sobre la necesidad e importancia de estudiar este tema. El 60% de los estudiantes indicaron que brindar conocimientos en esta área es necesario para la formación integral de la carrera.



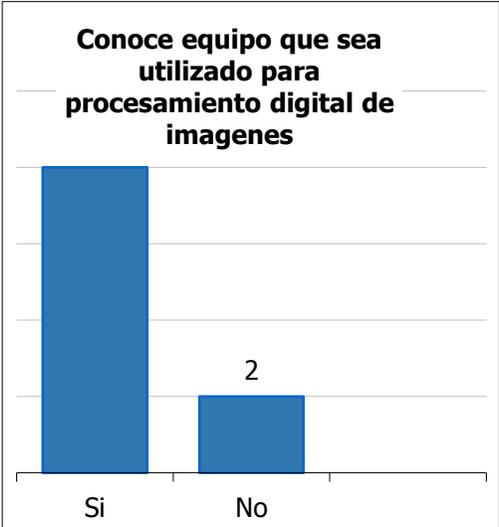
Grafica 2: Resultados de los estudiantes encuestados.

Se les consulto a los estudiantes si tenían conocimientos sobre el equipo que se propone utilizar el cual es la Raspberry Pi. Los resultados pueden apreciarse en la Grafica 3, de 10 estudiantes entrevistados el 60% respondió positivamente.



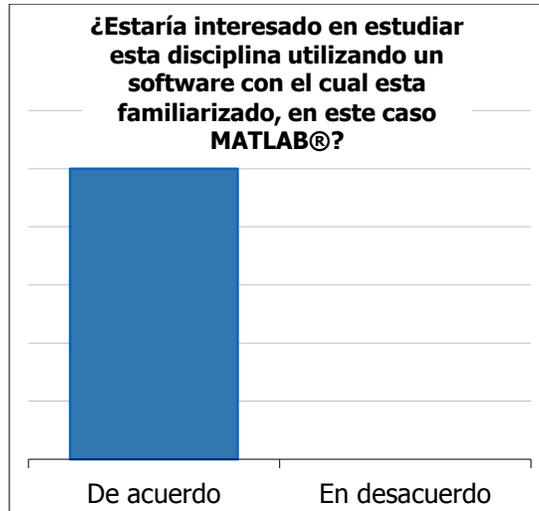
Grafica 3: Resultados estudiantes encuestados.

En el Grafico 4 se obtiene resultados positivos respecto al conocimiento de equipos dedicados a este tipo de procesamiento. Sin embargo, en la Grafica 5 vemos que en general los estudiantes concuerdan en que en la universidad no hay o no conocen la existencia de un equipo orientado a esta área.



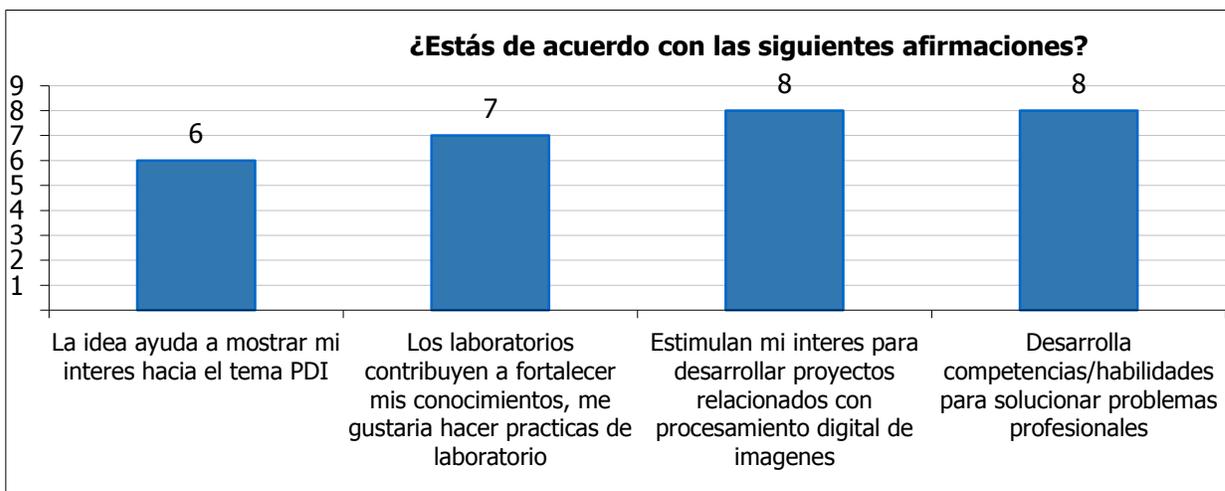
Grafica 4: Resultados estudiantes encuestados.

Generalmente en la Grafica 5 podemos observar que la propuesta hecha a los estudiantes de utilizar un software con el que están sumamente familiarizados tuvo un nivel de aceptación del 100%.



Grafica 5: Resultados estudiantes encuestados.

En el Grafico 6 podemos ver que un 60% de estudiantes se muestran interesados en la idea del Procesamiento Digital de Imágenes. Un 70 % piensa que los laboratorios en general ayudan al fortalecimiento de los conocimientos y por último, el 80% opina que este tipo de actividades estimulan el desarrollo de proyectos y habilidades para la resolución de situaciones en el ámbito profesional.



Grafica 6: Resultados estudiantes encuestados.

Como punto final de la encuesta se solicitó a los estudiantes que ofrecieran sugerencias de mejora hacia las prácticas de laboratorio, las cuales están reflejadas en el siguiente cuadro:

Sugerencias de mejora	
1	Los laboratorios deben estar mejor equipados más con equipos para experimentación, ya que los equipos utilizados ni siquiera eran de la facultad.
2	Sería una excelente idea que este tema se brinde en alguna asignatura como una unidad del programa de dicha asignatura. Esto nos permitiría profundizar más en el tema.
3	Investigar y tener a disposición periféricos que nos permitan utilizar en conjunto con la tarjeta de desarrollo Raspberry Pi
4	Más equipos de laboratorio para la realización de prácticas.
5	Indagar más a fondo en esta área, para así poder brindar temas para proyectos.

2.2.2 Entrevistas aplicadas a docentes.

Las entrevistas⁵ aplicadas fueron realizadas a docentes que han impartido asignaturas orientadas a Sistemas de Control, Procesamiento de Señales o asignaturas donde se involucre asignación de proyectos de final de curso. Esto con el fin de obtener una valoración del docente en la importancia o necesidad que existe para tener conocimientos en esta disciplina.

a) *Entrevista a Ing. Denis Vílchez*

De la entrevista realizada a Ing. Denis Vílchez se destaca que en la carrera de Ingeniería Electrónica, no se ha impartido ningún tema referente a procesamiento digital de imágenes.

⁵ En Anexo "A": Entrevista realizadas a docentes

También expresa que en la facultad no ha visto o manipulado ningún equipo que se utilice para este tema.

b) ***Entrevista a Ing. Martha Ligia González.***

El punto de vista expuesto en la entrevista realizada a la Ing. Martha Ligia nos da un punto de vista interesante. Una de las principales áreas de aplicación de esta disciplina como es la Medicina.

También expresa la falta de equipos y herramientas para experimentación en la universidad.

c) ***Entrevista a Ing. Carlos Ortega***

Con la entrevista obtenida con el Ing. Carlos Ortega tuvimos algunos datos distintos a los docentes anteriores, por ejemplo, le gusta la propuesta de trabajar esta área con MATLAB® sin embargo nos habla de la propuesta de trabajar con otros lenguajes como son Python y Open CV.

También nos habla de una posibilidad de que esta disciplina sea utilizada en dos empresas en sus sistemas de control de calidad.

Se diseñaron las guías de laboratorio en base al análisis de las encuestas aplicadas a los estudiantes y las entrevistas aplicadas a los docentes.

2.3 Diseño de guías de laboratorios

Para el diseño de las guías de laboratorio⁶ de Procesamiento Digital de Imágenes utilizando Raspberry Pi, se seleccionó esta tarjeta debido a que se puede obtener un módulo de cámara dedicado a ella. Una vez que se tuvo acceso a este equipo se inició a realizar distintos experimentos. Estos experimentos iniciales consistieron en primeramente crear una conexión desde la PC vía cable de red o de manera inalámbrica⁷. Luego se descargó el Toolbox correspondiente desde el software MATLAB®.

⁶ En Anexo "B": Guías de Laboratorio

⁷ En Anexo "C": Conexión Raspberry Pi

Seguidamente se empezaron a realizar experimentos de adquisición de imágenes en distintas resoluciones y con distintos filtros. Eligiendo esto como primer experimento adecuado se procedió al diseño de la guía de laboratorio.

Se diseñaron 4 guías de laboratorio que abarcan distintos algoritmos de aplicación en el procesamiento de imágenes. Estos algoritmos pueden ser utilizados en cualquier asignatura impartida a lo largo de la carrera.

Las guías de laboratorio se describen a continuación, así como una síntesis del contenido de la guía:

a) **Laboratorio 1:** Captura y pre-procesamiento de las imágenes

Para la primera guía de laboratorio no se solicitó trabajo previo ya que la idea de esta guía fue familiarizarse con el equipo de adquisición de imágenes propuesto. Utilizando la Raspberry Pi y el módulo de cámara, se enfocaron distintos objetivos (cuadernos, compañeros, etc.). Para cumplir con los objetivos establecidos. A lo largo de la guía se detalla todo lo esencial con respecto a MATLAB® y la Raspberry Pi incluyendo los módulos disponibles.

b) **Laboratorio 2:** Detección de los bordes de una imagen

En la segunda guía se estudió el algoritmo EDGE disponible en el software MATLAB®, este algoritmo nos permite detectar los bordes en una imagen. La guía se enfoca en utilizar primeramente el algoritmo sin ninguna variante, y una vez hecho eso hacer el análisis usando las distintas variantes de EDGE (Sobel, Prewitt, Canny, etc.).

c) **Laboratorio 3:** Detección de características en objetos

Como tercer laboratorio se propuso la implementación del algoritmo SURF, el cual consiste en la detección de puntos característicos en una imagen. Brindando el script correspondiente se les solicito a los estudiantes que enfocaran el módulo de cámara a un objetivo (botella, cuaderno, mochila, etc.) y aplicarles dicho algoritmo para luego analizar los resultados.

d) **Laboratorio 4:** Detección de rostros y cuerpos

En esta guía de laboratorio se comprobó la eficacia de los algoritmos de detección de rostros y detección de cuerpos en el cual los estudiantes modificaron el script dado para aplicar estos algoritmos a una secuencia de video.

La estructura de las guías de laboratorio consta de la siguiente manera:

- El **Tema** de la práctica de laboratorio.
- **Objetivos:** se plantean los logros que se persiguen con la práctica de laboratorio.
- **Introducción:** una breve reseña de lo que se planea estudiar en la guía de laboratorio.
- **Descripción:** en esta sección se encuentran los puntos a abordar a lo largo de las guías.
- **Desarrollando el código:** en este punto se estudian y ejecutan los códigos brindados en las guías de laboratorio.
- **Conclusiones.**
- **Informe:** se indica la forma en que debe entregarse los resultados de la práctica de laboratorio.

2.4 Implementación de las guías de laboratorio.

En este proyecto se contó con la participación de 10 estudiantes de la carrera de Ingeniería Electrónica e Ingeniería en Telecomunicaciones de entre las generaciones 2011 a 2013.

Debido a la cantidad de equipos con los que se contaban (uno propio y uno de un estudiante), se decidió dividir al grupo en dos para implementar las guías en distintos horarios pero garantizando que fueran el mismo día, de esta manera la evaluación sería general para los participantes.

En esta misma sección se presenta el desempeño de los estudiantes que consistió en identificar el comportamiento que presentaron durante el desarrollo de cada una de las guías de laboratorio correspondiente al contenido abordado en las guías. A

la compresión de cada una de las actividades descritas en la guía, al trabajo cooperativo, entre otros aspectos.

2.4.1 Laboratorio 1: Captura y pre-procesamiento de las imágenes

En este laboratorio se dedicaron aproximadamente 20 – 25 minutos para realizar una descripción del tema, lo que era Linux y también lo que era la Raspberry Pi, también se les facilitó a los estudiantes el documento correspondiente a la configuración de los modos de conexión de la Raspberry Pi con la computadora.

En esta práctica se logró observar claramente el interés de los estudiantes por el la disciplina de procesamiento de imágenes. También desconocían la apariencia física de la Raspberry Pi. Una vez iniciada la práctica no se presentó ningún inconveniente a la hora de ir realizando cada uno de los puntos establecidos en la guía.

2.4.2 Laboratorio 2: Detección de los bordes de una imagen

En este laboratorio se explicó en qué consistía el algoritmo EDGE para que los estudiantes tuvieran una idea más clara además de la brindada en el documento, se procedió a dar una breve reseña de las distintas variantes de este algoritmo disponibles en MATLAB®.

Una vez finalizada la explicación se procedió a realizar lo que el documento solicita, pero una vez llegados al segundo ejercicio los estudiantes empujados por la propia curiosidad probaron las distintas variantes orientando a diferentes objetivos.

Finalizando la guía de laboratorio el ejercicio tres no pudo completarse debido a la falta de tiempo.

2.4.3 Laboratorio 3: Detección de Objetos a través de características

Se les indicó a los estudiantes al leer la descripción de la guía que sustenta el fundamento teórico del algoritmo a estudiar. En este caso se utilizó el algoritmo SURF. Mediante la explicación del contenido de la guía de laboratorio se aclaró en qué consistía dicho algoritmo.

Una vez iniciada la práctica por los estudiantes se notó la actitud favorable en el desarrollo de la guía y el trabajo en equipo para obtener un objetivo adecuado según solicita la guía.

En el transcurso de la práctica de laboratorio se observó que los estudiantes desarrollaron bien el contenido de la guía, aprendieron a utilizar el algoritmo SURF y a diferenciar objetos distintos a través de sus características. En este laboratorio el tiempo fue suficiente para desarrollar toda la guía.

2.4.4 Laboratorio 4: Detección de rostros y cuerpos

La guía de laboratorio se les proporcionó el mismo día en que realizaron el laboratorio, debido a la imposibilidad de terminar el diseño de la guía antes del día establecido para el laboratorio.

La práctica de laboratorio fue realizada en pareja y se observó una actitud dinámica entre los estudiantes para la solución y comprensión de la guía de laboratorio. La intervención del instructor consistió en la revisión y recomendación de algunos puntos del trabajo de los estudiantes.

Finalizando la sesión de laboratorio la mayoría de los estudiantes lograron resolver la guía de laboratorio, mientras que otros presentaron dificultad en la solución de la guía entregándola incompleta. Sin embargo, se notó que poseían conocimientos de la plataforma de programación MATLAB®.

A continuación se sintetizan los resultados en base al desempeño observado en cada una de las prácticas con las guías de laboratorio implementadas a los estudiantes. Se estableció una escala de evaluación la cual consiste en Nulo (N), Regular (R), Muy Bueno (MB) y Excelente (E).

Parámetros	Evaluación
La explicación inicial del contenido del laboratorio contribuyo a un buen desempeño en el desarrollo del laboratorio.	E
Los estudiantes comprendieron los procedimientos indicados en la guía de laboratorio	MB
Presentaron una actitud dinámica en el desarrollo de la práctica.	MB
Se observó que los estudiantes adquirieron un nuevo conocimiento.	E
Comprendieron los resultados de la práctica de laboratorio.	B
Estaban atentos a la solución de las actividades propuestas en la guía de laboratorio.	MB
Se observó el trabajo en equipo.	E
Se cumplió con los objetivos establecidos en la guía de laboratorio.	B
Resolvieron 60% las actividades de la guía correctamente	MB
Resolvieron en un 90% las actividades de la guía correctamente.	E
La guía de laboratorio se ajustó al tiempo establecido para la sesión de laboratorio.	MB

2.5 Evaluación de resultados

2.5.1 Encuesta final aplicada a estudiantes con los que se implementaron las guías de laboratorio.

Para determinar la satisfacción del grupo de estudiantes con las guías de laboratorio desarrolladas para el Procesamiento Digital de Imágenes, se les aplicó una encuesta⁸ a cada estudiante tomando criterios que comprende el desempeño práctico de ellos, el aprendizaje adquirido, el contenido teórico de las guías, las limitaciones que se presentaron y las sugerencias por parte de los estudiantes.

Para representar el resultado de evaluación de cada uno de los criterios de evaluación se empleó el gráfico de barras.

⁸ En Anexo "D": Encuesta final aplicada a los estudiantes

Los criterios de evaluación que se tomaron en cuenta en la encuesta estaban conformado de la siguiente manera:

a) Nivel de desempeño: el desempeño práctico correspondió a la aceptación de los estudiantes en el desarrollo de todas las guías de laboratorio realizadas con el software y hardware disponible para la experimentación, su nivel de satisfacción muestra el interés que se obtuvo con las guías propuestas para las prácticas de laboratorios. Para evaluar el nivel de desempeño practico se utilizó la siguiente escala:

- Muy insatisfecho
- Insatisfecho
- Satisfecho
- Muy satisfecho

b) Nivel de aprendizaje: el nivel de aprendizaje está basado en la incidencia que tuvieron los contenidos de cada práctica de laboratorio para obtener otros conocimientos además de los estudiados en las sesiones de clases. Por consiguiente se tomaron cuatro niveles de evaluación que determinan el aprendizaje adquirido en los laboratorios. El nivel aprendizaje corresponde a:

- Irrelevante
- Poco significativo
- Significativo
- Muy significativo

c) Nivel de contenido teórico: este criterio se tomó en cuenta a partir de los temas desarrollados en cada una de las guías de laboratorio. El nivel teórico consistió en determinar qué tan conveniente fue utilizar los contenidos presentados en las guías de laboratorio. La escala de avaluación están dadas por:

- Inoportuno
- Poco oportuno

- Oportuno
- Muy oportuno

En ese mismo nivel de evaluación se determinó el alcance que tuvo el contenido teórico en el aprendizaje de los estudiantes por cada uno de las guías de laboratorio desarrolladas: estableciendo los niveles de evaluación “Nada”, “Poco”, “Necesario” y “Suficiente”, indicando con escala del 1 al 4 respectivamente en un diagrama de barras.

En una escala de cero a cien por ciento de los conocimientos a aprender en el curso, se toma como “Poco” el nivel donde el estudiante aprendió dicho conocimiento entre un cero y un cincuenta y nueve por ciento; como “Necesario” el nivel donde el estudiante aprendió dicho conocimiento entre sesenta y un setenta y nueve por ciento y “Suficiente” el nivel donde el estudiante aprendió dicho conocimiento entre un ochenta y un cien por ciento.

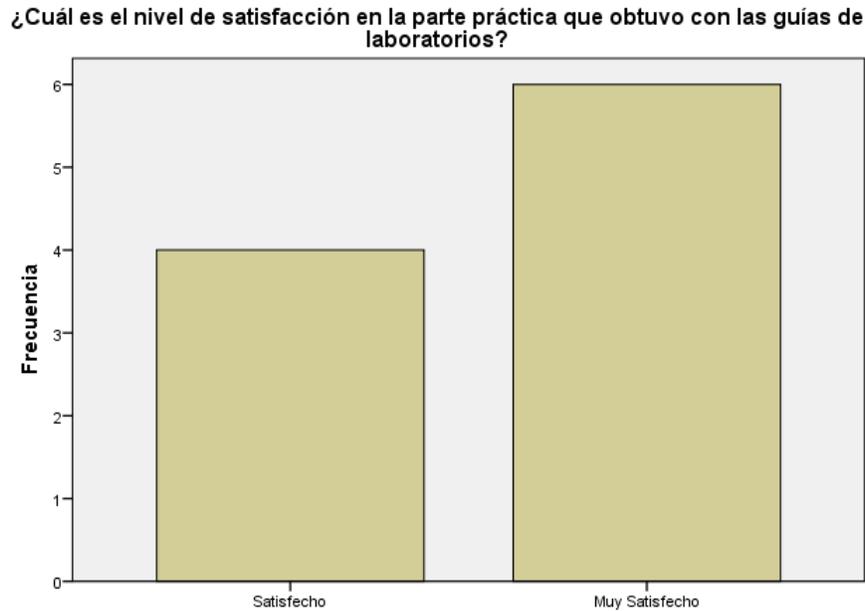
d) Limitaciones: la sección comprendía aquellas limitaciones que afectaron el desarrollo satisfactorio de las guías de laboratorio, estas incluyen la cantidad de tiempo destinada para la práctica, la utilidad del contenido y los materiales empleados para realizar las sesiones de laboratorio de montaje.

La respuesta a cada una de estas limitaciones de este apartado están dadas por las opciones “Si” y “No”, representadas en el grafico 4 por un 1 para “Si” y un 2 para “No”.

2.6 Resultados de las encuestas finales aplicadas a los estudiantes

a. Nivel de desempeño practico

El resultado del Grafico 8 muestra que los estudiantes se sintieron “Satisfechos” con el desarrollo de las guías de laboratorio.



Grafica 8: Resultados del nivel de desempeño practico

La tabla muestra de forma numérica los resultados obtenidos en el desempeño de los estudiantes respecto a las guías.

Tabla 1. ¿Cuál es el nivel de satisfacción en la parte práctica que obtuvo con las guías de laboratorios?

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido Satisfecho	4	40.0	40.0	40.0
Muy Satisfecho	6	60.0	60.0	100.0
Total	10	100.0	100.0	

Fuente: Encuestas aplicadas a estudiantes

La tabla muestra la cantidad porcentual de los estudiantes que destacaron su nivel de satisfacción con las prácticas realizadas donde el 60% de los estudiantes

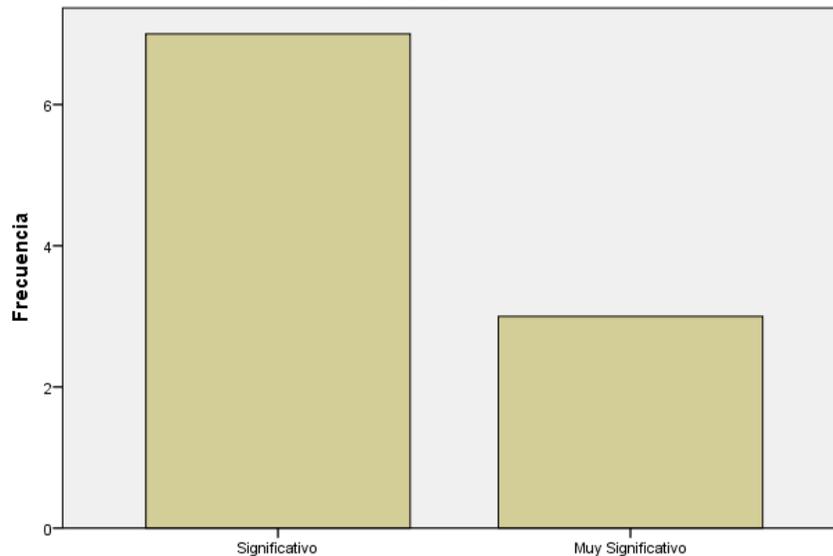
indicaron que están “Muy Satisfechos” con las guías desarrolladas, obteniendo una valoración positiva de todos los estudiantes.

Por lo tanto, las guías de laboratorio fueron bien aceptadas por más de la mayoría de los estudiantes en cada una de las sesiones de laboratorio, lo cual indica que los estudiantes mostraron un desempeño positivo ante las guías de laboratorio.

b. Nivel de aprendizaje

El Grafico 9 muestra que el aprendizaje obtenido por los estudiantes respecto al uso del módulo de cámara para la Raspberry Pi fue significativo, lo que indica la comprensión de todo lo necesario respecto a este componente.

¿Cómo fue el aprendizaje obtenido en los laboratorios acerca del uso de módulo de cámara?



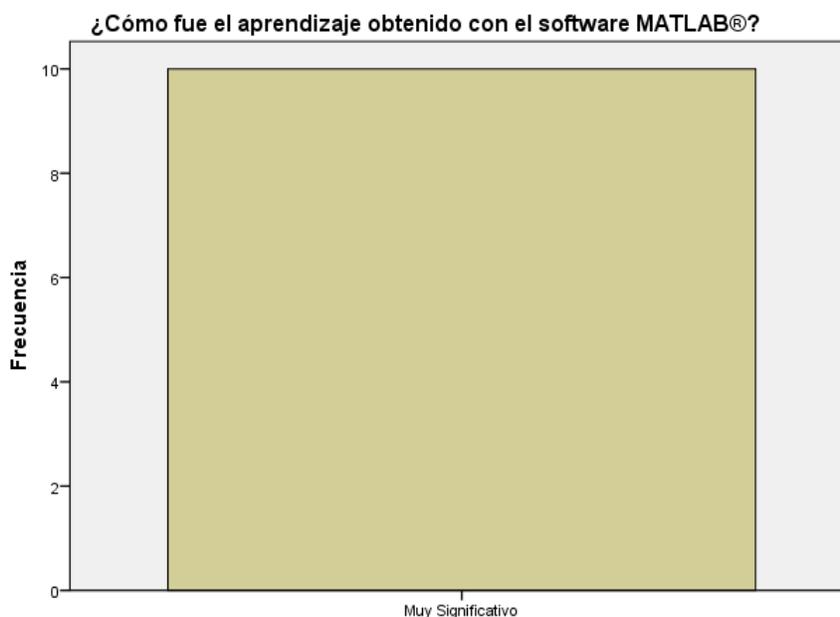
Grafica 9: Resultados del nivel de aprendizaje

En la Tabla 2 encontramos más claramente los resultados obtenidos del nivel de aprendizaje de los estudiantes.

Tabla 2. ¿Cómo fue el aprendizaje obtenido en los laboratorios acerca del uso de módulo de cámara?

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Significativo	7	70.0	70.0	70.0
	Muy Significativo	3	30.0	30.0	100.0
	Total	10	100.0	100.0	

La Tabla 2 refleja la cantidad porcentual de los estudiantes que destacan su nivel de aprendizaje obtenido en la manipulación del módulo de cámara. El resultado refleja que el 70% de los estudiantes indicaron que obtuvieron un aprendizaje “Significativo” sobre las características y uso de este módulo. La valoración positiva de esta encuesta fue del 100%.



Grafica 10: Resultados del nivel de aprendizaje

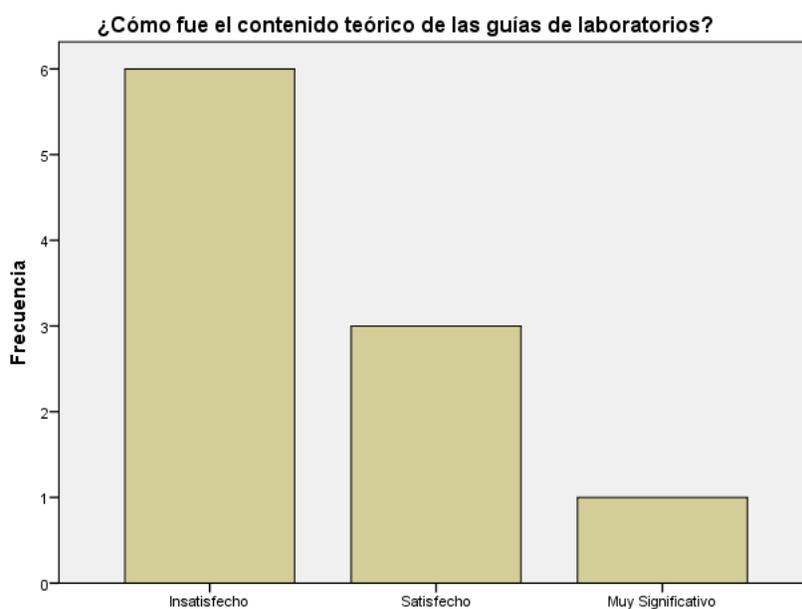
En el Grafico 10 podemos observar que el aprendizaje obtenido utilizando el software MATLAB® fue “Muy Significativo” para todos los estudiantes. El 100% de los estudiantes destacaron el nivel de satisfacción que tuvieron al utilizar este software.

En base a los resultados anteriores se identificó que las guías implementadas contribuyeron al desarrollo de un aprendizaje significativo en el uso y manipulación

del componente de la Raspberry Pi, también en el uso del software MATLAB® ya que como se ha hecho gran énfasis es utilizado a lo largo de la carrera y los estudiantes están familiarizados con este software.

c. Nivel de contenido teórico

De acuerdo a la tabla 3, se muestra que el 60% de los estudiantes no estaban contentos con el contenido teórico de las guías de laboratorio, “Insatisfecho” fue la opción más seleccionada y es debido a que en las guías de laboratorio no estaban diseñadas para brindar más conocimientos que los necesarios.



Grafica 11: Resultados del nivel contenido teórico

El alcance que tuvo el contenido teórico no fue suficiente para el 60% de los estudiantes, pero 30% de los estudiantes “Satisfechos” concluyen que con el objetivo de las guías es suficiente, esto debido a que en ninguna asignatura está previsto abarcar estos temas.

En las Tablas 3 a 7 se detalla el nivel de aprendizaje teórico obtenido en cada guía de laboratorio.

Tabla 3: Laboratorio No 1: Captura y pre-procesamiento de las imágenes.

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Poco	4	40.0	40.0	40.0
	Necesario	4	40.0	40.0	80.0
	Suficiente	2	20.0	20.0	100.0
	Total	10	100.0	100.0	

En la Tabla 3 se detalla que el 40% de los estudiantes opinaron que aprendizaje teórico recibido fue “Poco” mientras que el otro 40% opinaron que fue “Necesario”, el resto tuvieron una opinión positiva por lo que se puede concluir que el primer laboratorio abarcó un contenido teórico suficiente para ser un documento introductorio.

Tabla 4: Laboratorio No 2: Detección de los bordes de una imagen.

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Poco	5	50.0	50.0	50.0
	Suficiente	5	50.0	50.0	100.0
	Total	10	100.0	100.0	

En la Tabla 4 un 50% de los estudiantes opinaron que fue “Poco” el contenido teórico obtenido de la guía mientras que el otro 50% opina que fue “Suficiente” el contenido teórico aprendido en la guía.

Tabla 5: Laboratorio No 3: Detección de Objetos a través de características.

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Poco	6	60.0	60.0	60.0
	Necesario	4	40.0	40.0	100.0
	Total	10	100.0	100.0	

La Tabla 5 no muestra que el 60% de los estudiantes tuvieron una adquisición de contenido teórico baja en el laboratorio número tres el cual se enfoca en el algoritmo EDGE. Mientras que el resto apenas considera que fue “Necesaria”.

Tabla 6: Laboratorio No 4: Detección de rostros y cuerpos.

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Poco	8	80.0	80.0	80.0
	Suficiente	2	20.0	20.0	100.0
Total		10	100.0	100.0	

Para finalizar esta parte la tabla 5 nos muestra como el 80% sintieron que el contenido teórico abarcado en la guía numero 4 no fue suficiente.

En conclusión, en esta sección podemos decir que los estudiantes no se sintieron satisfechos con el contenido teórico abarcado en las guías de laboratorio.

d. Limitaciones

Las limitaciones están enfocadas a tres principales puntos, estos fueron el tiempo, el contenido teórico abordado y la disponibilidad de los equipos y materiales para realizar las prácticas.

Tabla 7: Resumen Limitaciones

Valido Si No	¿Cree que el tiempo fue suficiente para desarrollar satisfactoriamente el laboratorio?		¿Fue útil el contenido teórico abordado en cada una de las guías de laboratorio?		¿Es necesario utilizar más equipos para desarrollar las prácticas?	
	Si	No	Si	No	Si	No
Resultados	10	0	10	0	10	0

En la tabla 7 podemos analizar que el 100% de los estudiantes opinan que se deben obtener mayor cantidad de equipos ya que con los que se estuvo trabajando son muy pocos, también todos opinaron que el contenido teórico abordado fue útil y necesario para ampliar sus conocimientos y por último el 100% también opino que una hora-clase fue suficiente para realizar las practicas.

La importancia de conocer las sugerencias u opiniones de los estudiantes refleja la satisfacción que tuvieron y sus expectativas posteriores a las prácticas de laboratorio; de tal forma que dan un vistazo de las posibles mejoras que ellos mismos recomiendan.

Numero De Encuesta	Sugerencias
1	<ul style="list-style-type: none"> - Mejorar la base teórica - Si es posible más tiempo en los laboratorios
3	Más tiempo, más recursos y más laboratorios.
4	Dentro de las recomendaciones diría que un poco más de tiempo con mayores revisiones del tutor.
5	Más graficas e imágenes que sean explícitas sobre los temas a abordar en las prácticas de laboratorio.
7	Aumentar el número de laboratorios y la parte teórica.
8	Mejores y más equipos.
9	Ir aumentando poco a poco la complejidad y el reto que estas guías representan.

CAPITULO III. CONCLUSIONES Y RECOMENDACIONES

3.1 Conclusiones

En las encuestas aplicadas a los estudiantes involucrados en este trabajo se reflejó que poseían conocimientos básicos del software MATLAB® pero ninguno tenía conocimientos en el área de Procesamiento Digital de imágenes.

Se diseñaron cinco guías de laboratorios, las cuales todas fueron enfocadas a trabajar desde la interfaz de MATLAB®, utilizando la plataforma Raspberry Pi en conjunto con su módulo de cámara como tarjeta de adquisidor. Un alto porcentaje de los estudiantes señalaron que los contenidos fueron de gran interés.

Se aplicaron las guías de laboratorios a un grupo de estudiantes de entre las carreras de Ingeniería Electrónica e Ingeniería en Telecomunicaciones. Resulto que la mayoría de los estudiantes participaron de forma activa al momento en que se desarrollaban las prácticas de laboratorio, la mayoría también se vio involucrado en ambientes de enseñanza-aprendizaje promovido por el tutor.

Se aplicó una encuesta final a los estudiantes y resulto que el 80% de los estudiantes estuvieron satisfechos con las guías de laboratorios ya que el procesamiento se realizaba en tiempo real de la siguiente manera: adquisición del objetivo y procesamiento. El aprendizaje obtenido por los estudiantes fue significativo para comprender los temas de Procesamiento Digital de Imágenes.

3.2 Recomendaciones

- Se sugiere que algunos de los laboratorios de la Facultad sean equipados con más equipos orientados a esta disciplina ya sean Raspberrys Pi, módulos de cámara estándar, módulos de cámara Infrarroja, etc.
- Se recomienda trabajar con una versión de MATLAB® actual (2014 en adelante) ya que esta cuenta con los Toolbox necesarios para la Raspberry Pi.
- Las guías presentadas en este trabajo monográfico pueden mejorarse si se cuenta con los dispositivos necesarios mencionados anteriormente para el desarrollo de más laboratorios orientados a esta disciplina.

REFERENCIAS

- Boullosa, O. (2011). Estudio comparativo de descriptores visuales para la detección de escenas Cuasi-Duplicadas.
- Buemi, M. E. (2012). *Tratamiento de imágenes de radar de apertura sintética mediante filtros stack*. Buenos Aires.
- Delgado Montiel, J. L. (2010). *Reconocimiento de Placas Vehiculares*. México D.F.
- Diaz, I., Prieto, F., & Branch, J. (n.d.). *Segmentación de imágenes de rango por detección de bordes*. Colombia.
- Evans, C. (2009). Notes on the OpenSURF Library. 25.
- Faúdez. (2012). Tratamiento Digital. In M. Faudez, *Tratamiento Digital de voz e imagen y aplicación a la multimedia*. Alfa Omega.
- Foundation, R. P. (2015). *Raspberrypi*. Retrieved from <https://www.raspberrypi.org/about/>
- Gutiérrez Reyes, J. J., & Mejía Hernández, I. R. (2009). *Diseño e implementación de medios de laboratorio para contribuir al proceso de enseñanza en el área de automatización industrial*.
- Herbert Bay, L. G., & Tuytelaars, T. (n.d.). SURF: Speeded Up Robust Features.
- Jonassen, D. H. (1991). *Evaluating constructivistic learning*.
- Kindsein. (2007, 03 03). *Alzheimer*. Retrieved from <http://www.kindsein.com/es/18/alzheimer/423/?tpl=37>
- Mathwork. (2015). *MATLAB*. Retrieved from <http://www.mathworks.com/products/matlab/>
- Nazareth, Á. R. (2013). Retrieved from http://visualmed.usal.es/?page_id=162
- Pi, R. (2015, 2). *Camera Module*. Retrieved from <https://www.raspberrypi.org/products/>
- Pi, R. (2015, 2). *Power*. Retrieved from <https://www.raspberrypi.org/help/faqs/#power>
- Pina, A. N. (2010). Clasificación y búsqueda de imágenes usando características visuales.
- Pregunton. (2014). *www.pregunton.org*. Retrieved from <http://www.pregunton.org/pregunta.php?id=1019>
- Raspberry Pi Foundation. (n.d.). *raspberrypi*. Retrieved from <https://www.raspberrypi.org/about/>
- Requena, S. H. (2008). El modelo constructivista con las nuevas tecnologías aplicado en el proceso de aprendizaje. *Revista de Universidad y Sociedad del Conocimiento*, 10.
- RPiHardware. (2015, 5 30). *elinux*. Retrieved from <http://elinux.org/>
- Sean, L. (2015). *www.nuzart.com*. Retrieved from <http://www.nuzart.com/blog/2013/12/13/las-fotografias-infrarrojas-de-las-montanas-de-nepal/>

ANEXOS

ANEXO A

Entrevistas realizadas a docentes

Entrevista.

Esta pequeña entrevista es dirigida a docentes que han impartido asignaturas de asignaturas de Sistemas de Control, Procesamiento de Señales o asignaturas donde se involucre asignación de proyectos de final de curso, con el fin de determinar el desempeño de las guías de laboratorios en los estudiantes, la aceptación que causo en los estudiantes y las habilidades de aprendizaje que obtuvieron durante el desarrollo de las prácticas de laboratorio en dichas asignaturas; a como también la valoración del docente hacia las guías utilizadas.

Nombre del entrevistado: **Denis Vílchez**

- 1) En el tiempo que tiene de impartir clases, ¿Recibió o impartió algún tema referente al procesamiento digital de imágenes? *No, debido a que desde que ingrese como estudiante en esta carrera el pensum y sus modificaciones no han contemplado este tema.*
- 2) ¿Cree usted que sería de utilidad tener conocimientos en esta área para la carrera de Ingeniería Electrónica? *Claro que si porque actualmente es parte de la tecnología que encontramos a diario en todos lados, desde nuestros teléfonos celulares hasta nuestros televisores y computadoras, etc.*
- 3) Sabemos que MATLAB® es una de las herramientas software más utilizadas por los estudiantes, ¿Cuál es su opinión referente al desarrollo de medios de laboratorio en el área de procesamiento de imágenes utilizando esta herramienta? *MATLAB es una herramienta poderosa no que es utilizada para muchas asignaturas y pienso que ya que este software tiene ciertas facilidades que nos pueden ayudar en esta área.*
- 4) ¿Tiene conocimiento de que exista en la Facultad de Electrotecnia y Computación equipo hardware dedicado a esta disciplina? *No.*



- 5) ¿Conoce empresas o alguna industria que implemente esta tecnología en sus procesos? *Si, canales de televisión, Ineter, etc.*

Entrevista.

Esta pequeña entrevista es dirigida a docentes que han impartido asignaturas de asignaturas de Sistemas de Control, Procesamiento de Señales o asignaturas donde se involucre asignación de proyectos de final de curso, con el fin de determinar el desempeño de las guías de laboratorios en los estudiantes, la aceptación que causo en los estudiantes y las habilidades de aprendizaje que obtuvieron durante el desarrollo de las prácticas de laboratorio en dichas asignaturas; a como también la valoración del docente hacia las guías utilizadas.

Nombre del entrevistado: **Ing. Martha Ligia**

- 1) En el tiempo que tiene de impartir clases, ¿Recibió o impartió algún tema referente al procesamiento digital de imágenes? Explique.

No, en realidad no estaba contemplado en el pensum

- 2) ¿Cree usted que sería de utilidad tener conocimientos en esta área para la carrera de Ingeniería Electrónica? ¿Por qué?

Claro que sí, es un campo más de la electrónica el cual tienen amplias aplicaciones en la medicina.

- 3) Sabemos que MATLAB® es una de las herramientas software más utilizadas por los estudiantes, ¿Cuál es su opinión referente al desarrollo de medios de laboratorio en el área de procesamiento de imágenes utilizando esta herramienta?

Sería muy conveniente, creo que hasta cuenta con algunos Toolbox para esta área.

- 4) ¿Tiene conocimiento de que exista en la Facultad de Electrotecnia y Computación equipo dedicado (hardware) a esta disciplina?

No, desconozco eso.

5) ¿Conoce empresas o alguna industria que implemente esta tecnología en sus procesos?

Principalmente podría hablar por los hospitales que es donde se aprecia de mejor manera esta área, en sus aparatos electromédicos.

Entrevista.

Esta pequeña entrevista es dirigida a docentes que han impartido asignaturas de asignaturas de Sistemas de Control, Procesamiento de Señales o asignaturas donde se involucre asignación de proyectos de final de curso, con el fin de determinar el desempeño de las guías de laboratorios en los estudiantes, la aceptación que causo en los estudiantes y las habilidades de aprendizaje que obtuvieron durante el desarrollo de las prácticas de laboratorio en dichas asignaturas; a como también la valoración del docente hacia las guías utilizadas.

Nombre del entrevistado: **Ing. Carlos Ortega**

- 1) En el tiempo que tiene de impartir clases, ¿Recibió o impartió algún tema referente al procesamiento digital de imágenes? Explique.

No, cuando fui estudiante no tuvimos ninguna alusión al tema y en mi tiempo como docente no se ha contemplado este tema.

- 2) ¿Cree usted que sería de utilidad tener conocimientos en esta área para la carrera de Ingeniería Electrónica? ¿Por qué?

Nosotros como ingenieros debemos ser integrales, tener conocimientos en todas las áreas posibles.

- 3) Sabemos que MATLAB® es una de las herramientas software más utilizadas por los estudiantes, ¿Cuál es su opinión referente al desarrollo de medios de laboratorio en el área de procesamiento de imágenes utilizando esta herramienta?

Siendo este software de gran trascendencia en la carrera, sería muy conveniente. Sin embargo hay otras herramientas útiles como Python, OpenCV, etc. Las cuales pueden ser beneficiosas para esta área.

- 4) ¿Tiene conocimiento de que exista en la Facultad de Electrotecnia y Computación equipo dedicado (hardware) a esta disciplina?

Sé que en el laboratorio de automatización existe un equipo que debería ser utilizado para eso sin embargo nunca lo he visto en operación.

- 5) ¿Conoce empresas o alguna industria que implemente esta tecnología en sus procesos?

E escuchado que utilizan equipos que usan algunos métodos de detección en la Cervecera Nacional de Nicaragua y en la planta de la Coca-Cola.

ANEXO B

Guías de Laboratorio

Laboratorio 1: Captura y pre-procesamiento de las imágenes

1. Objetivos

- Familiarizarse con el software MATLAB® y la Raspberry Pi.
- Manejar los pines de propósito general GPIO.

2. Introducción

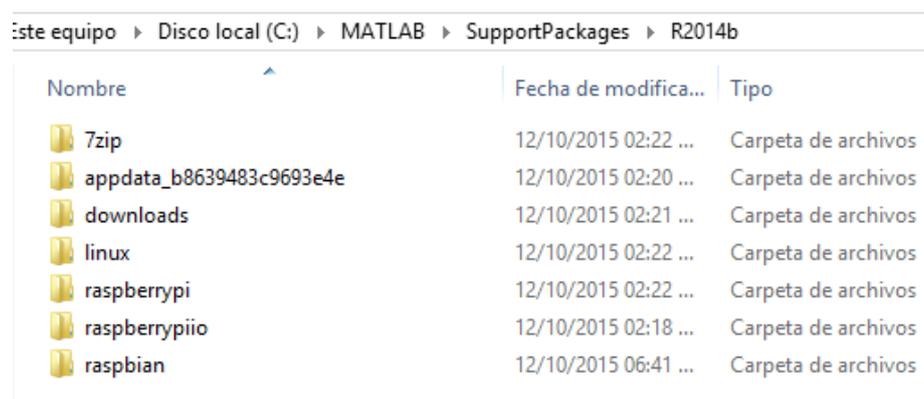
El procesamiento de imágenes mediante MATLAB® consiste principalmente en obtener una imagen y poder reconocer aspectos de ella y tratarlos. Para poder realizar dichas tareas tenemos que modificar la imagen original de diferentes formas normalmente siguiendo un esquema general.

Esta guía de laboratorio se encargada de obtener una imagen, leerla y prepararla para el análisis que se le quiera hacer posteriormente, la cual consiste en explicar las operaciones de lectura, cambios de color y de escalas de grises, filtrado del ruido de la imagen, etc. que se le realizan a una imagen para poder analizarla correctamente.

3. MATLAB®

MATLAB® es un software matemático el cual nos ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (Lenguaje M). Disponible para todas las plataformas Unix, Windows y Mac OS. Entre sus principales prestaciones encontramos la manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaces de usuario (GUI) y la comunicación.

En la locación de la carpeta MATLAB®, usualmente en “C:\C:\MATLAB\SupportPackages\Version”, están las carpetas con los archivos del Toolbox descargados para MATLAB®. Ver **Figura 1**



Nombre	Fecha de modifica...	Tipo
7zip	12/10/2015 02:22 ...	Carpeta de archivos
appdata_b8639483c9693e4e	12/10/2015 02:20 ...	Carpeta de archivos
downloads	12/10/2015 02:21 ...	Carpeta de archivos
linux	12/10/2015 02:22 ...	Carpeta de archivos
raspberrypi	12/10/2015 02:22 ...	Carpeta de archivos
raspberrypiio	12/10/2015 02:18 ...	Carpeta de archivos
raspbian	12/10/2015 06:41 ...	Carpeta de archivos

Figura 1. Carpeta *SupportPackages*

En esta carpeta estarán todos los datos necesarios para que MATLAB® trabaje con la Raspberry.

Tipos numéricos:

Descripción			
Nombre	Tamaño	Rango	Descripción
<i>double</i>	8 bytes	[0, 1]	Números de precisión doble en el rango aproximado de -10^{308} a 10^{308} . Si es una imagen con valores escalados el rango es el antes mencionado
<i>uint8</i>	1 byte	[0, 255]	Enteros sin signo de 8-bits en el rango
<i>uint16</i>	2 bytes	[0, 65535]	Enteros sin signo de 16-bits en el rango
<i>uint32</i>	4 bytes	[0, 4294967295]	Enteros sin signo de 32-bits en el rango
<i>int8</i>	1 byte	[-128, 127]	Enteros con signo de 8-bits en el rango
<i>int16</i>	2 bytes	[-32768, 32767]	Enteros con signo de 16-bits en el rango
<i>int32</i>	4 bytes	[-2147483648, 2147483647]	Enteros con signo de 32-bits en el rango
<i>single</i>	4 bytes	$[-10^{38}, 10^{38}]$	Números de precisión simple en el rango aproximado de -10^{38} a 10^{38}
<i>char</i>	2 bytes	N/A	Caracteres
<i>logical</i>	1 byte	N/A	Valores de 0 o 1

4. Raspberry Pi

Raspberry Pi es un ordenador que tiene un tamaño tan pequeño como el de una tarjeta de crédito. En la siguiente tabla podemos ver un resumen de las principales características de cada modelo:

Raspberry Pi	Generación 1				Generación 2
	Modelo A	Modelo B	Modelo A+	Modelo B+	Modelo 2 B
Alimentación	300mA	700mA	200mA	600mA	900mA
Puerto Ethernet	No	Si	No	Si	Si
Puertos USB	1	2	1	4	4
GPIO	26	26	40	40	40
Puerto SD	SD	SD	microSD	microSD	microSD
SoC	BCM2835	BCM2835	BCM2835	BCM2835	BCM2835
CPU	700MHz ARM11 Single-core	700MHz ARM11 Single-core	700MHz ARM11 Single-core	700MHz ARM11 Single-core	900MHz ARM Cortex-A7 Quad-core
RAM	256MB	512MB	256MB	512MB	1GB

En las serie de guías de laboratorio propuestas se hará uso del modelo Raspberry PI 2 B, esto debido a sus prestaciones de hardware y que en el mercado es la principal disponible.

Componentes Básicos

El diagrama de la Raspberry Pi 2 B nos muestra los componentes básicos que encontramos en la tarjeta. Ver **Figura 2**

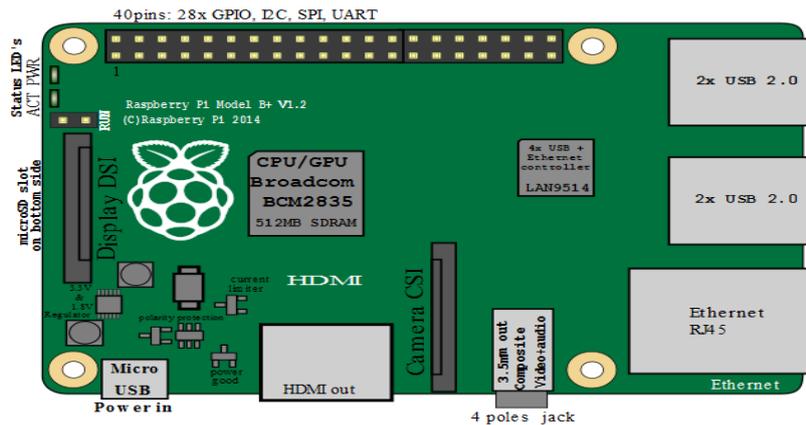


Figura 2. Raspberry Pi 2B

- **Conector de alimentación:** la tarjeta es alimentada con 5 V los cuales son suministrados por un puerto USB Micro.
- **Puerto HDMI (High-Definition Multimedia Interface):** es la conexión para pantalla de Alta Definición (HD) ya sean monitores o televisores LCD.
- **Conector para interfaz de cámara serial (CSI):** este es un conector para un módulo de cámara.
- **Conector de audio:** es una salida de audio a través de un Jack de 3.55 mm que también soporta.
- **Puerto Ethernet:** es un conector de red que brinda velocidades de 10/100 Mbits/s a través de un cable RJ45.
- **Puerto USB:** con este puerto tenemos compatibilidad con cualquier dispositivo como teclados, mouse, Wi-Fi dongle, o cámara WEB.
- **GPIO:** es un set de pines de entrada y salida de propósito general, por ejemplo tarjetas de expansión, dispositivos de control del CPU. La segunda generación tienen 40 pines.
- **LED's:** sirven para indicar el estado de la tarjeta, por ejemplo, cuando está alimentada el LED rojo se enciende y si está conectada a una red el LED amarillo se activa.
- **Display Serial Interface (DSI):** este conector sirve para conectar un display, como el PiTFT, el cual es una pantalla de cristal líquido (Thin Film Transistor).
- **Conector micro SD-Card:** sirve para insertar una memoria micro-SD que almacena el Sistema Operativo (OS).

5. Procedimiento

5.1 Crear un script en MATLAB®

- a. Abrir el MATLAB®. Ver **Figura 3**



Figura 3. Logo de MATLAB®

- b. Dar click en el botón **New Script** para crear in “archivo.m”. Ver **Figura 4**

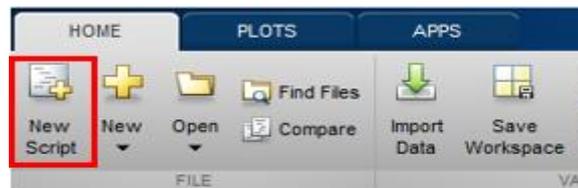


Figura 4. Inicio en la ventana principal de MATLAB®.

- c. Copiar los códigos de la Tabla 1 del inciso 5 en MATLAB® como se muestra a continuación. Ver **Figura 5**

```
Editor - C:\Users\Lab-Digital\Documents\MATLAB\Lab1.m
Lab1.m x +
1 - clear %Limpia las variables guardadas en MATLAB
2 - rpi = raspi(); %Crea una conexión con la Raspberry Pi
3 - cam = cameraboard(rpi, 'Resolution', '800x600');%Crea una conexión con el
4 %Modulo de Camara
5
6 %% Importa y muestra una secuencia de 100 capturas en la computadora
7 - for ii = 1:100
8 | img = snapshot(mycam)
9 | imagesc(img)
10 | drawnow
11 - end
```

Figura 5. Código de Tabla 1

- d. Una vez copiado el código, dar clic en la pestaña “**EDITOR**” (Ver **Figura 6**) y luego presionar el botón “**Run**” (Ver **Figura 7**).



Figura 6. Pestaña **EDITOR**

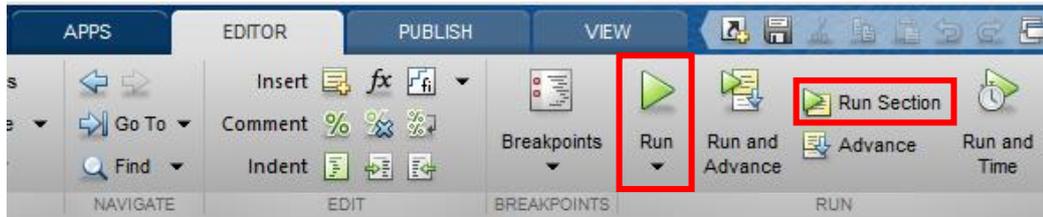


Figura 7. Botón *RUN* y *Run Section*

El botón *RUN* nos permite ejecutar el script completo mientras que el botón *Run Section* solo ejecutara una sección del script a la vez. Las secciones están separadas por “%%”.

6. Desarrollando el Código

Hardware:

Este laboratorio mostrará cómo manipular los pines GPIO y el módulo de cámara de la Raspberry. En este caso, la Raspberry también cuenta con un Led integrado que podemos programar.

Las principales funciones que tenemos disponibles en la línea de comando del MATLAB® son:

- **Raspi:** esta función crea una conexión con la tarjeta.
- **Cameraboard:** crea una conexión con el módulo de cámara de la Raspberry.
- **Snapshot:** captura una imagen RGB desde el módulo de cámara.
- **Record:** captura video desde el módulo de cámara.
- **configureDigitalPin:** configura un pin específico del GPIO solo salida digital o entrada.
- **readDigitalPin:** lee el valor lógico de un pin GPIO.
- **writeDigitalPin:** escribe un valor lógico en un pin del GPIO.
- **writeLED:** coloca un valor lógico en el led integrado en la tarjeta.
- **showLEDs:** muestra la localización del led integrado en la tarjeta.

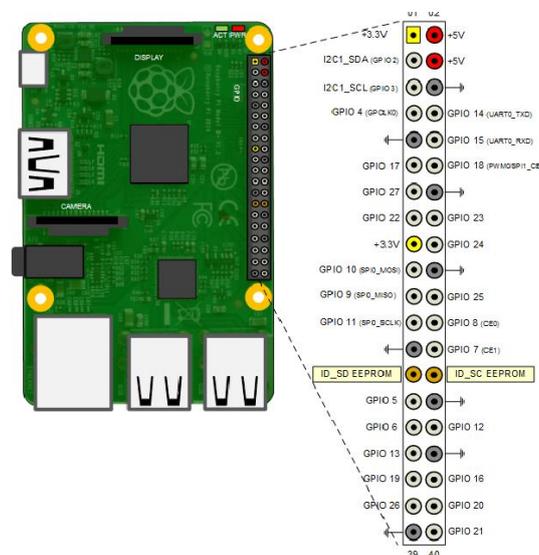


Figura 8. Descripción del GPIO

Software:

Teniendo conocimiento de las funciones que nos presenta el toolbox de Raspberry en MATLAB®, debemos tener en cuenta que ayuda ofrece información y ejemplos de las funciones del toolbox. Ver **Figura 9**

```
for ii = 1:100
    buttonPressed = readDigitalPin(mypi,23)
    if buttonPressed
        for jj = 1:10
            writeDigitalPin(mypi,24,1)
            pause(0.05)
            writeDigitalPin(mypi,24,0)
            pause(0.05)
        end
    end
    pause(0.1)
end
```

Figura 92. Funciones de entrada/salida en MATLAB®, para controlar los pines GPIO.

Programación:

Siempre es conveniente recordar que disponemos de la ayuda del MATLAB® para cualquier duda respecto a cualquier comando:

LED_Blink y Snapshot

El siguiente código sirve para hacer parpadear el LED integrado en la tarjeta cada 0.5 segundo y también hacer capturas desde la cámara en la Raspberry. Ver **Tabla 1**

```
clear %Limpia las variables guardadas en MATLAB
rpi = raspi(); %Crea una conexión con la Raspberry Pi
cam = cameraboard(rpi, 'Resolution', '800x600');%Crea una conexión con el
%Módulo de Cámara

%% Importa y muestra una secuencia de 100 capturas en la computadora
writeLED(rpi, 'led0', false);

for i = 1:100
    img = snapshot(cam);
    imagesc(img);
    drawnow;
    writeLED(rpi, 'led0', 1);
    pause(0.05);
    writeLED(rpi, 'led0', 0);
    pause(0.05);
end
```

Tabla 1. Código LED_Blink1 y Snapshot

Aplicando distintos Filtros

El siguiente código sirve para tener ver el resultado al aplicar distintos filtros en la imagen capturada y aplicar diferentes pines GPIO:

```
clear
rpi = raspi();
cam = cameraboard(rpi, 'Resolution', '800x600');

%%Aplicar distintos filtros a las imagenes capturadas
cam.ImageEffect = 'negative'; % El comando "cam" nos permite ver las
for i = 1:50 % propiedades del módulo de cámara
    img = snapshot(cam);
    imagesc(img);
    drawnow;
    subplot(221);
    writeDigitalPin(rpi,24,1)
    pause(0.05)
    writeDigitalPin(rpi,24,0)
    pause(0.05)
end

cam.ImageEffect = 'sketch';
for i = 1:50
    img = snapshot(cam);
    imagesc(img);
    drawnow;
    subplot(222);
    writeDigitalPin(rpi,25,1)
    pause(0.05)
    writeDigitalPin(rpi,25,0)
    pause(0.05)
end

cam.ImageEffect = 'washedout';
for i = 1:50
    img = snapshot(cam);
    imagesc(img);
    drawnow;
    subplot(223);
    writeDigitalPin(rpi,23,1)
    pause(0.05)
    writeDigitalPin(rpi,23,0)
    pause(0.05)
end

cam.ImageEffect = 'oilpaint';
for i = 1:50
    img = snapshot(cam);
    imagesc(img);
    drawnow;
    subplot(224);
    writeDigitalPin(rpi,6,1)
    pause(0.05)
    writeDigitalPin(rpi,6,0)
    pause(0.05)
end
```

Tabla 2. Código LED_Blink2

7. Circuitos a montar

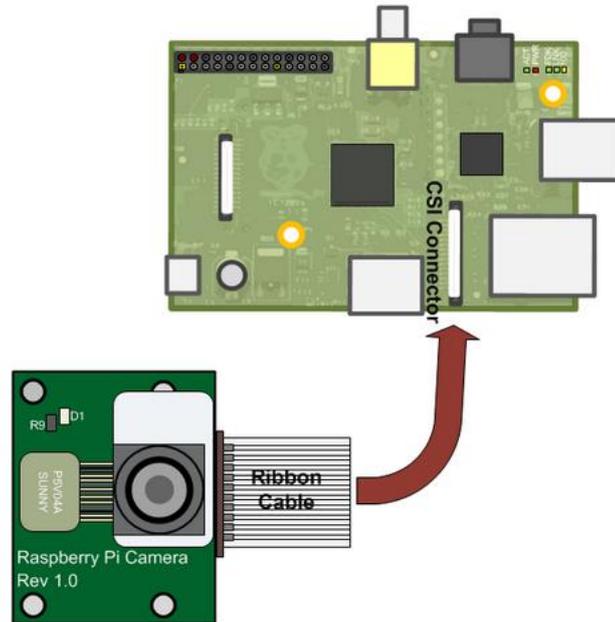


Figura 3. Circuito a simular para el código LED_Blink1

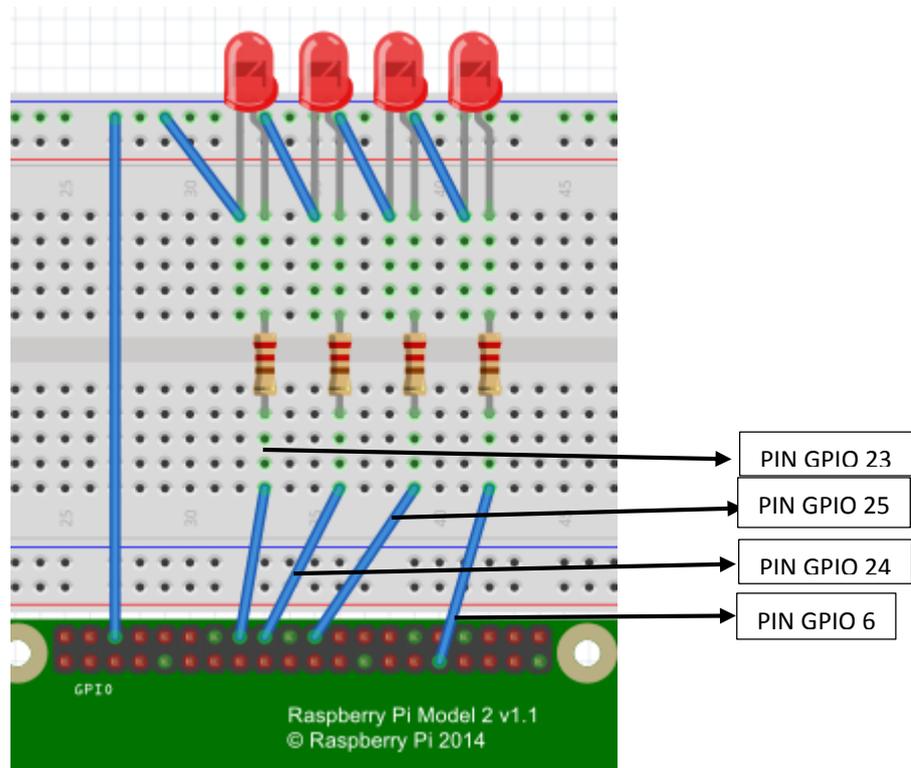


Figura 4. Circuito a simular para con el segundo código

Laboratorio 2: Detección de los bordes de una imagen

1. Objetivos

- Hacer uso de la Raspberry Pi para adquirir imágenes en tiempo real y aplicar los filtros correspondientes para la detección de bordes.
- Utilizar distintas mascarar empleadas para la detección de bordes en una imagen.

2. Introducción

La detección de bordes es una técnica de procesamiento de imágenes para la búsqueda de los límites de objetos dentro de las imágenes. Funciona mediante la detección de discontinuidades en el brillo. La detección de bordes se utiliza para la segmentación de imágenes y extracción de datos en áreas como procesamiento de imágenes, visión por computadora, y la visión de máquina.

Se lleva a cabo mediante derivadas de primer y segundo orden y la IPT cuenta con una función específica para ello, la función **edge**, desde la que se puede detectar los bordes de una imagen utilizando diferentes métodos.

3. Descripción

BW = edge(I) recibe como entrada una imagen binaria o en escala de grises *I* y devuelve una imagen binaria **BW** del mismo tamaño que *I*, con 1's donde la función haya encontrado bordes y 0's en el resto. *I* es un arreglo disperso tipo **numeric**. **BW** es de tipo **logical**.

edge utiliza por defecto el método **Sobel** para detectar los bordes, pero se pueden utilizar otros métodos. Algunos se listan a continuación:

- **Sobel**: detecta los bordes utilizando la aproximación **Sobel**. Suele tener buenos resultados con las opciones por defecto.
- **Roberts**: uno de los métodos más simples y antiguos de detección de bordes. No suele utilizarse mucho salvo que se necesite simpleza y rapidez.
- **Prewitt**: suele obtener resultados parecidos a los obtenidos con el método Sobel.

- **Canny:** detecta los bordes máximos locales del gradiente de I . Este método es el más potente de todos y por ello es el más utilizado. Es menos propenso a confundirse por el ruido y más propenso a detectar los bordes débiles. Para obtener el mejor resultado normalmente se necesitan modificar sus parámetros por defecto.

4. Desarrollando el código

Ejercicio #1

En el primer ejercicio se hará la adquisición de una imagen desde la Raspberry Pi y posteriormente pasarla a escala de grises y por ultimo haremos uso del comando **'edge'** para detectar los bordes de la imagen en blanco y negro y la representamos.

Crear un script con el nombre **"Lab_2A"**, y compilar el siguiente código:

```
%% Inicializacion de la Raspberry
clear                               %Limpia las variables guardadas en MATLAB
rpi = raspi();                       %Crea una conexión con la Raspberry Pi
cam = cameraboard(rpi, 'Resolution', '800x600'); %Crea una conexión con el
                                         %Módulo de Cámara

%% Captura y guardado
for i = 1:20
    img = snapshot(cam);
    imwrite(img, 'C:captura.jpg', 'jpg');
end

%% Aplicando el Filtro Edge
i = imread('captura.jpg');
I = rgb2gray(img);
DB = edge(I);
subplot(1,3,1);
imshow(i);
subplot(1,3,2);
imshow(I);
subplot(1,3,3);
imshow(DB);
```

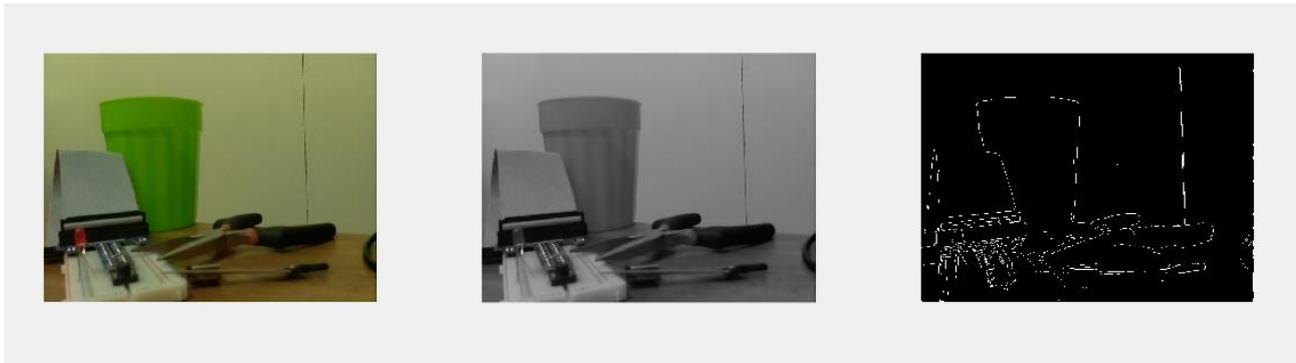


Figura 1. De izquierda a derecha: Captura original, escala de grises y filtro EDGE

Ejercicio #2

Para este ejercicio utilizaremos variantes de la función *'edge'* para observar la diferencia entre las imágenes al aplicar los diferentes métodos disponibles en la función:

Crear un script con el nombre **"Lab_2B"**, y compilar el siguiente código:

```
%% Inicializacion de la Raspberry
clear                               %Limpia las variables guardadas en MATLAB
rpi = raspi();                       %Crea una conexión con la Raspberry Pi
cam = cameraboard(rpi, 'Resolution', '800x600'); %Crea una conexión con el
                                         %Módulo de Cámara

%% Captura y guardado
for i = 1:20
    img = snapshot(cam);
    imwrite(img, 'C:captural.jpg', 'jpg');
end

%% Aplicando distintos metodos de edge
I = rgb2gray(img);

BW1 = edge(I, 'prewitt');
BW2 = edge(I, 'sobel');
BW3 = edge(I, 'roberts');

subplot(2,2,1); % Imagen original
imshow(I);      % en escala de grises
title('original');

subplot(2,2,2); % Aplicando filtro
imshow(BW1);   % Prewitt
title('Prewitt');

subplot(2,2,3); % Aplicando filtro
imshow(BW2);   % Sobel
title('Sobel');

subplot(2,2,4); %Aplicando filtro
imshow(BW3);   % Roberts
title('Roberts');
```

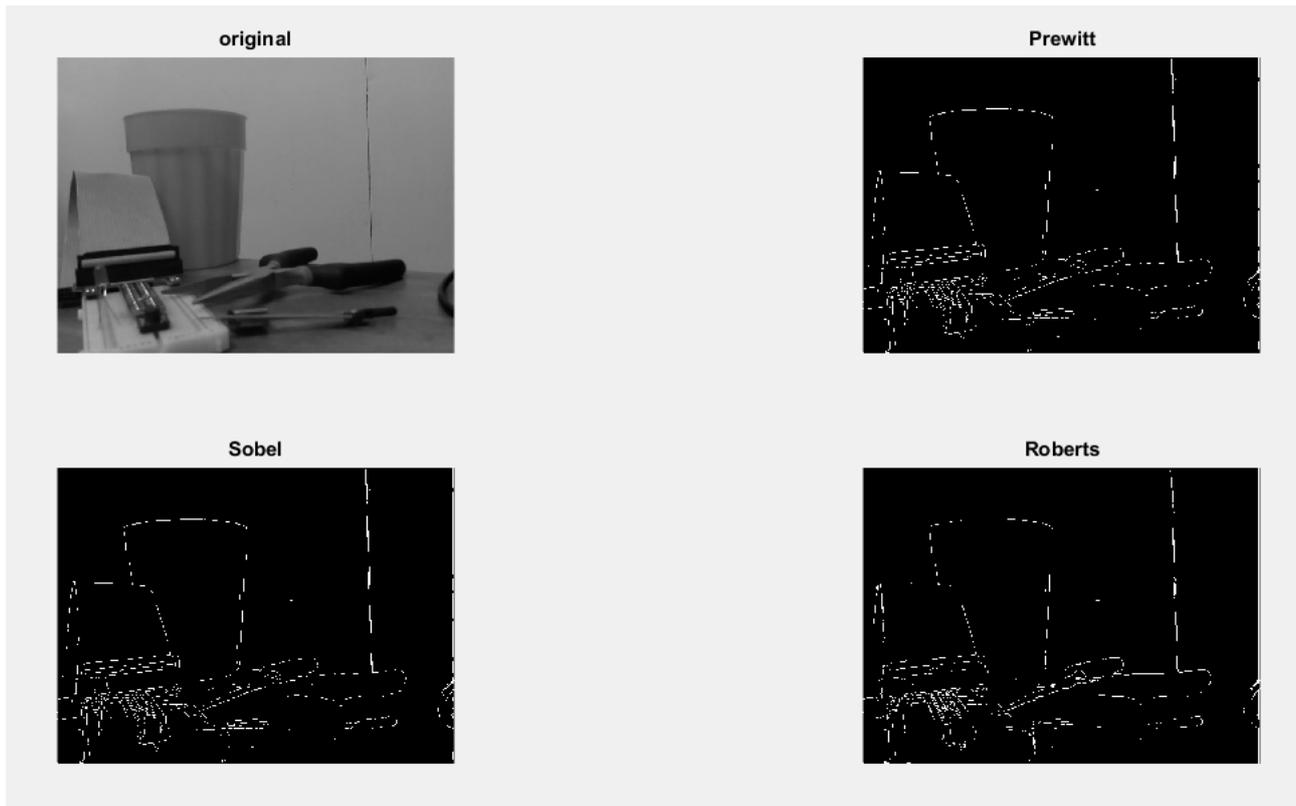


Figura2. Variantes del Filtro Edge

Hacer captura de una imagen lo suficientemente clara para poder apreciar la diferencia entre los diferentes filtros. Seguir probando el resto de variantes disponibles de la función **'edge'**.

Ejercicio #3

Para el tercer ejercicio el estudiante deberá investigar un método que permita detectar los bordes verticales en una imagen y también los bordes horizontales, utilizando siempre la función **'edge'** y cualquiera de sus variantes.

Mostrar el resultado al instructor una vez finalizado

Laboratorio 3: Detección de características en objetos

1. Objetivos

- Utilizar la Raspberry Pi para hacer captura de imágenes y utilizarlas de referencia para su posterior procesamiento
- Usar las características SURF que representan un objeto
- Detectar características desde dos fuentes

2. Introducción

SURF (speed up robust feature) es un detector es otro detector de variables locales, y fue presentado por primera vez por Herbet Bay en el 2006 y se inspira en el descriptor SIFT, pero presentando ciertas mejoras, como:

- Velocidad de cálculo considerablemente superior sin ocasionar perdida del rendimiento.
- Mayor robustez ante posibles transformaciones de la imagen.

Estas mejoras se consiguen mediante la reducción de la dimensionalidad y complejidad en el cálculo de los vectores de características de los puntos de interés obtenidos, mientras continúan siendo suficientemente característicos e igualmente repetitivos.

3. Descripción

SURF es otro de los algoritmos más utilizado para la extracción de puntos de interés en el reconocimiento de imágenes. La extracción de los puntos la realiza detectando en primer lugar los posibles puntos de interés y su localización dentro de la imagen.

Es mucho más rápido que el método SIFT, ya que los keypoints contienen muchos menos descriptores debido a que la mayor cantidad de los descriptores son 0.

Este descriptor se puede considerar una mejora debido a que las modificaciones que supondría en el código no serían excesivas, ya que el descriptor SURF utiliza la gran mayoría de las funciones que utiliza el descriptor SIFT.

A continuación se describe cual es el procedimiento del algoritmo SURF para detectar

puntos de interés (keypoints), asignación de la orientación y por ultimo obtención del descriptor SURF.

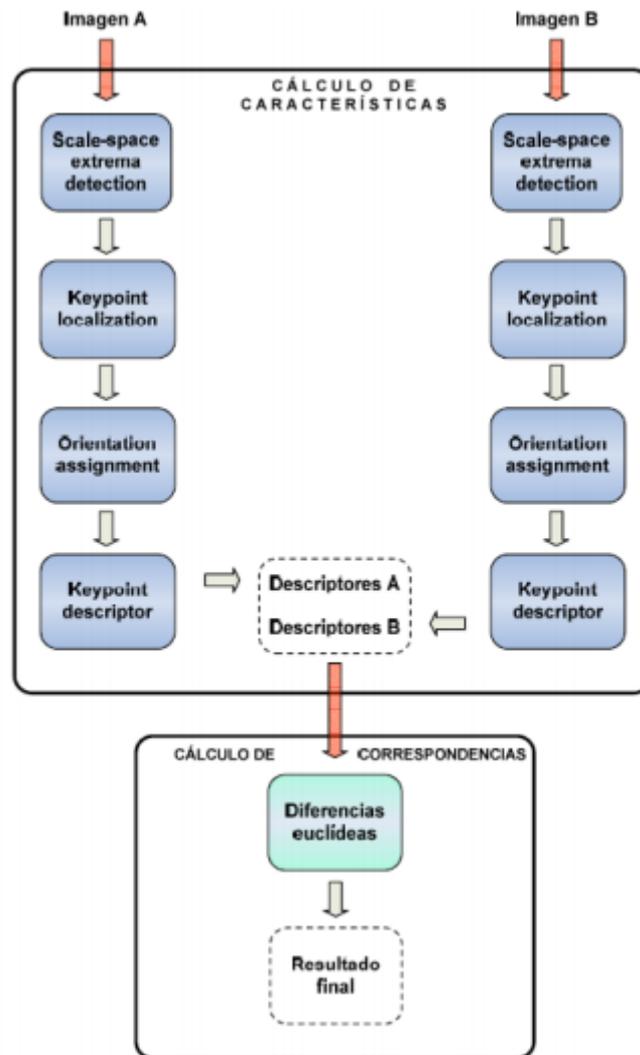


Figura 1: Esquema del proceso para la clasificación SURF

4. MATLAB®

Ejercicio #1

En este primer ejercicio, consiste en identificar las características de una imagen haciendo uso del algoritmo SURF. Para ello deberemos hacer captura de una imagen que contenga un objeto (una botella, portada de un cuaderno, etc.) y luego recortarla con la herramienta Paint.



Figura 2: Ejemplo de objeto seleccionado

Utilizando el siguiente script nos permitirá capturar la imagen y guardarla en el directorio de MATLAB® para luego extraer las características principales y luego mostrarlas.

Crear un script con el nombre “Lab_3”, y compilar el siguiente código:

```
clear                                %Limpia las variables guardadas en MATLAB
rpi = raspi();                        %Crea una conexión con la Raspberry Pi
cam = cameraboard(rpi, 'Resolution', '1024x768'); %Crea una conexión con el
                                         %Módulo de Cámara

%Captura de la Imagen
for i = 1:20
    img = snapshot(cam);
    imwrite(img, 'C:captura.jpg', 'jpg');
end

%% Carga la imagen de referencia (captura.png), y computa las características
SURF
ref_img = imread('captura1.png');
ref_img_gray = rgb2gray(ref_img);
ref_pts = detectSURFFeatures(ref_img_gray);
[ref_features, ref_validPts] = extractFeatures(ref_img_gray, ref_pts);

figure; imshow(ref_img);
hold on; plot(ref_pts.selectStrongest(50));
```



Figura 3: Principales características de la imagen Captura

Ejercicio #2

Agregar el segundo script nos permitirá identificar y visualizar las 25 características más representativas del objeto el cual está guardado con el nombre de **“Captura.jpg”** en el directorio de MATLAB®.

El segundo código nos permite extraer y visualizar algunas características seleccionadas por el algoritmo SURF.

```
%% Visualizacion de 25 caracteristicas
figure;
subplot(5,5,3); title('Primeras 25 Caracteristicas');
for i=1:25
    scale = ref_pts(i).Scale;
    image = imcrop(ref_img,[ref_pts(i).Location-10*scale 20*scale 20*scale]);
    subplot(5,5,i);
    imshow(image);
    hold on;
    rectangle('Position',[5*scale 5*scale 10*scale
10*scale], 'Curvature',1, 'EdgeColor','g');
end
```

Ejercicio #3

Ahora extraeremos las características de la imagen original utilizando la misma técnica que el Ejercicio #1. Una vez ejecutado el script de abajo podremos visualizar ciertas características en la imagen original.

```
% Características de la captura original  
image = imread('Original.jpg');  
I = rgb2gray(image);  
  
% Detectar características  
I_pts = detectSURFFeatures(I);  
[I_features, I_validPts] = extractFeatures(I, I_pts);  
figure;imshow(image);  
hold on; plot(I_pts.selectStrongest(50));
```



Figura 4: Imagen Original aplicando SURF

Ejercicio #4

Ahora que ya tenemos las características de cada imagen (Imagen de referencia y Captura), lo que haremos será comparar las características encontradas en cada imagen.

```
% Comparar imagen Original con Imagen de Referencia  
index_pairs = matchFeatures(ref_features, I_features);  
  
ref_matched_pts = ref_validPts(index_pairs(:,1)).Location;  
I_matched_pts = I_validPts(index_pairs(:,2)).Location;  
  
figure, showMatchedFeatures(image, ref_img, I_matched_pts, ref_matched_pts,  
'montage');  
title('Mostrar las Coincidencias');
```

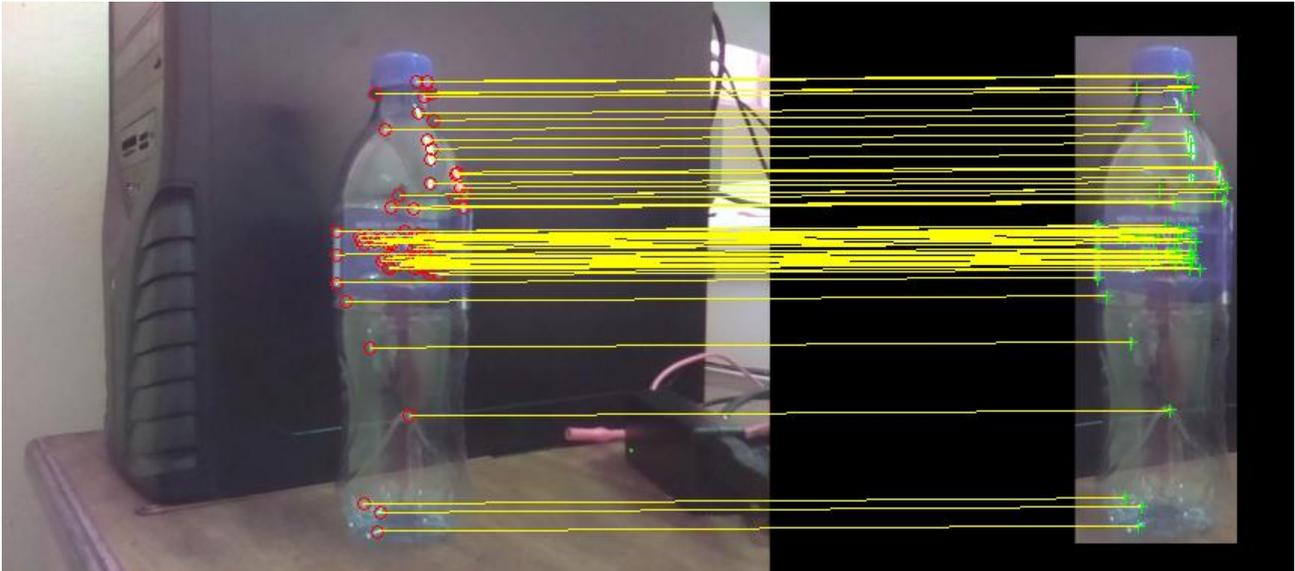


Figura 5: Comparación de Características

Mostrar el resultado al instructor una vez finalizado

Laboratorio 4: Detección de rostros y cuerpos

1. Objetivos

- Utilizar la Raspberry Pi para hacer captura de imágenes y utilizarlas de referencia para su posterior procesamiento
- Usar script para la detección de rostros
- Usar script para la detección de cuerpos

2. Introducción

El People Detector Object detecta personas en una imagen de entrada utilizando Histogramas de Gradientes Orientados (HOG) y también Máquina de vectores de soporte (SVM). El objeto detecta personas que no estén en una posición vertical.

Descripción

Los pasos para detectar personas son los siguientes:

- Define y setea el objeto detector de personas usando el constructor.
- Llama el método con una imagen de entrada, y el objeto detector de personas, `l`, `peopleDetector`. Para esto ver las sintaxis para el uso de estos métodos.

`BBOXES = step(peopleDetector,l)` realiza una detección multiescala de objetos en la imagen entrada, `l`. el método regresa una matriz de `M-4` definiendo `M` cuadros delimitadores, donde `M` representa el número de personas detectadas.

`[BBOXES, SCORES] = step(peopleDetector,l)` retorna un valor seguro para las detecciones. `M-1` vectores, `SCORES`, contiene valores positivos para cada delimitador en la `BBOXES`. Un largo record de valores indica un mayor valor seguro en la detección.

3. MATLAB®

Ejercicio #1

El primer ejercicio consiste en realizar la captura de una persona, para esto deberemos enfocar la Raspberry Pi en uno de nuestros compañeros y ejecutar el siguiente script.

```
%% Preparamos la captura con nuestra Raspberry Pi
clear
mypi = raspi();
myCam = cameraboard(mypi, 'Resolution', '640x480');

mySnap = snapshot(myCam);
imshow(mySnap);
hold on
```

Una vez realizado esto la siguiente sección del script se encargara de detectar una persona en posición vertical.

```
%% Detectar personas en posicion vertical
peopleDetector = vision.PeopleDetector;
[bboxes, scores] = step(peopleDetector, mySnap);
I_people = insertObjectAnnotation(mySnap, 'rectangle', bboxes, scores);
figure, imshow(I_people);
```

Ahora se procederá a probar otro modelo de detección.

```
%% Probar otro modelo
peopleDetector =
vision.PeopleDetector('ClassificationModel', 'UprightPeople_128x64');
peopleDetector.WindowStride = [4 4];
peopleDetector.MinSize = [256 128];
[bboxes, scores] = step(peopleDetector, I);
I_people = insertObjectAnnotation(I, 'rectangle', bboxes, scores);
figure, imshow(I_people);
```

Ejercicio #2

Para el segundo ejercicio los script que se brindan son utilizados para la detección de rostros.

```
%% Detect Face and insert a rectangle around the face

fD = vision.CascadeObjectDetector();
bbox = step(fD, mySnap);
imageOut = insertObjectAnnotation(mySnap, 'rectangle',bbox, 'Face');
imshow(imageOut);
title('Detected face');

%% Repeat in a loop to make it look like face detection in video
someBigNumber = 1000;
flag = 1;

for i = 1:someBigNumber
    ONorOFF = readDigitalPin(mypi,23);
    if ONorOFF == 1 % ON, Push button is pressed
        clearvars -except mypi myCam flag
        mySnap = snapshot(myCam);
        imshow(mySnap);
        hold on
        fD = vision.CascadeObjectDetector();
        bbox = step(fD, mySnap);
        imageOut = insertObjectAnnotation(mySnap, 'rectangle',bbox, 'Face');
        imshow(imageOut);
        title('Detected face');
        drawnow
    else % OFF, Push button is NOT pressed
        % Do nothing!
    end
end
clear
clc
```

Trabajo en casa

1 – Utilizar los scripts del Ejercicio #1 para realizar detección de personas en una secuencia de video.

2 – Realizar pruebas de detección de un único rostro entre un grupo de personas y también experimentar cuanto permite el método girar una selección.

ANEXO C

Conexión Raspberry Pi

Conexión del Raspberry Pi directamente al PC usando cable de red

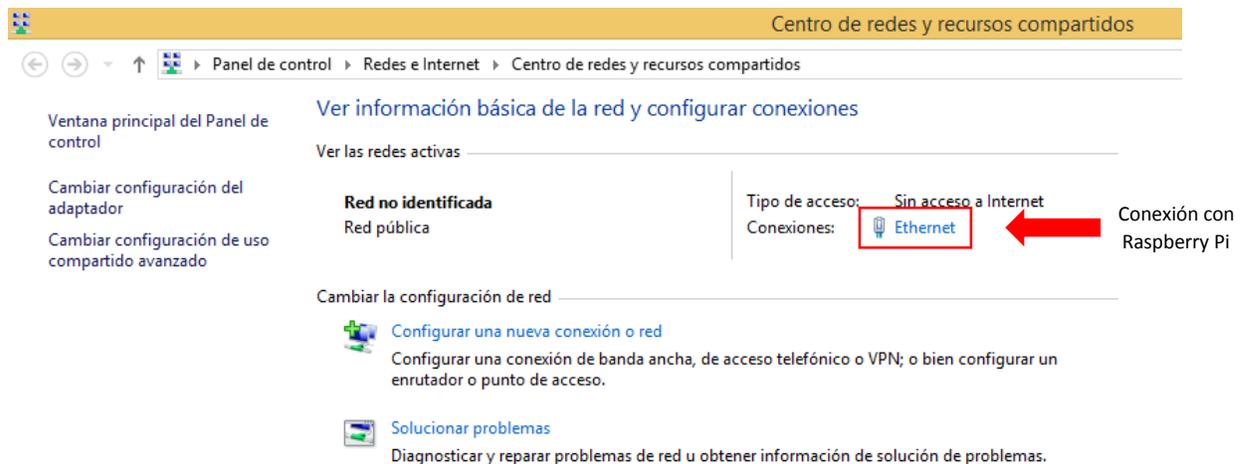
Podemos conectar directamente la Raspberry Pi a su PC utilizando un cable de red RJ-45. La velocidad de conexión es de hasta 100 Mbps. Las tarjetas de red de los nuevos portátiles y la del Raspberry Pi son Auto-Sense (Auto-MDI) y selecciona la interfaz de manera automática.

Configuración de Windows 7 o 8

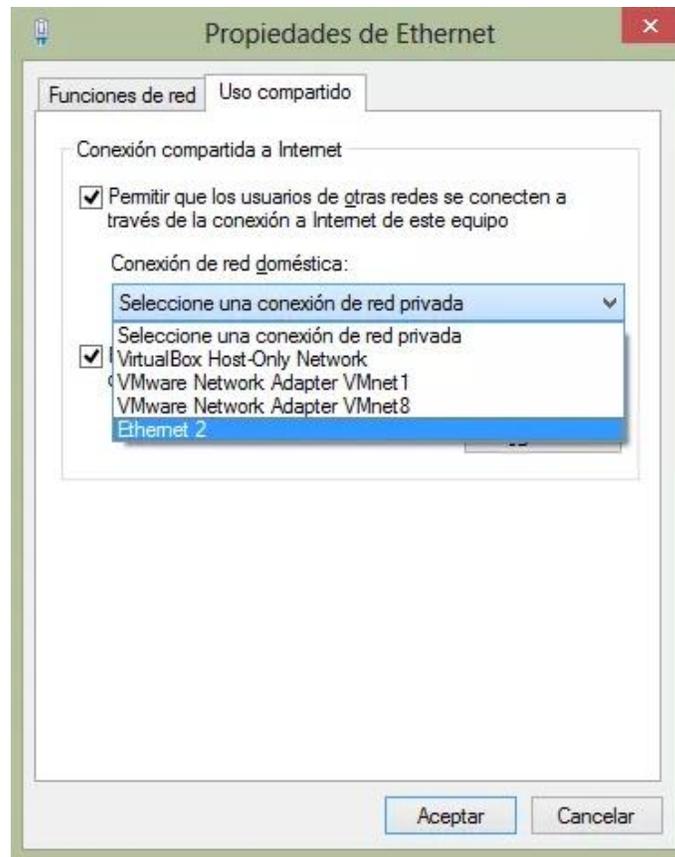
Conectar el cable RJ-45, las luces en la tarjeta de red se deben encender en ambos extremos, tanto en la Raspberry Pi como en la PC.

Ya que lo que necesitamos es que nuestro Raspberry Pi se conecte con nuestro computador, debemos realizar las siguientes configuraciones en la PC con Windows 7 u 8.1:

Paso 1. Es necesario compartir la conexión Wi-Fi o la conexión Ethernet que usted usa para acceder a la red, para esto es necesario activar es uso compartido de Internet entre las dos interfaces disponibles, haga clic derecho en el icono de red ( o ). Luego haga clic en “Abrir el Centro de redes y recursos compartidos”.

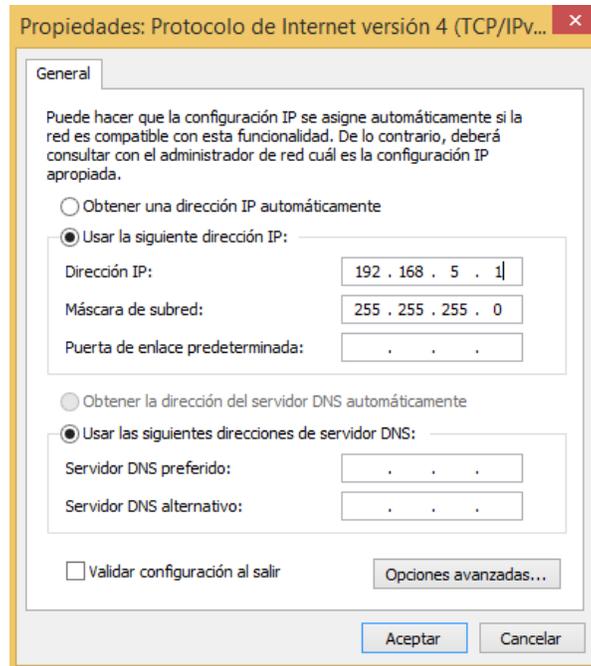


Paso 2. Identifique y seleccione su conexión primaria a Internet y haga clic sobre “*Ethernet*”. Esto abrirá la ventana de “*Estado de Ethernet*”. Haga clic en el botón  Propiedades y seleccione en el pestaña llamada “*Uso compartido*” y active la selección “*Permitir que los usuarios de otras redes se conecten a través de la conexión a Internet de este equipo*”. Luego en la caja de selección busque la conexión de red que tiene su RaspPi que debe ser la puerto Ethernet que está libre.



Paso 3. Configuramos la dirección estática de su PC en el puerto en el que se encuentra conectado la Raspberry Pi.

Repita el paso 1, ahora seleccione el puerto donde su Raspberry Pi está conectado, en el caso de este ejemplo es el puerto "Ethernet". Hacemos clic en el botón "**Propiedades**", luego en la caja "**Esta selección usa los siguientes elementos**" busque la sección "**Protocolo de Internet versión 4**", seleccione este campo y luego clic en propiedades. Para el caso de nuestro ejemplo, se va a asignar la red 192.168.5.1 a este puerto de conexión, verifique que todo está de acuerdo a la siguiente imagen.



Configurar dirección estática del Raspberry Pi

Para configurar un IP estática en la Raspberry debemos ingresar a la línea de comando de Linux, una vez allí, ingresamos en siguiente comando, el cual nos abrirá una ventana de edición para configurar la dirección estática de la Raspberry Pi:

```
sudo nano /etc/network/interfaces
```

Paso 1. Una vez que abrimos el archivo debemos modificar la tercera columna *“dhcp”* por la palabra *“static”* y asignamos la dirección IP 192.168.5.2 a la Raspberry Pi. Para que se pueda establecer la conexión entre la PC y RaspPi, la configuración deberá quedar de acuerdo a la siguiente imagen.

Una vez que hemos modificado el archivo presionamos *“Ctrl+X”* y luego *“S”* y Enter para guardar las modificaciones que le hemos hecho al archivo.

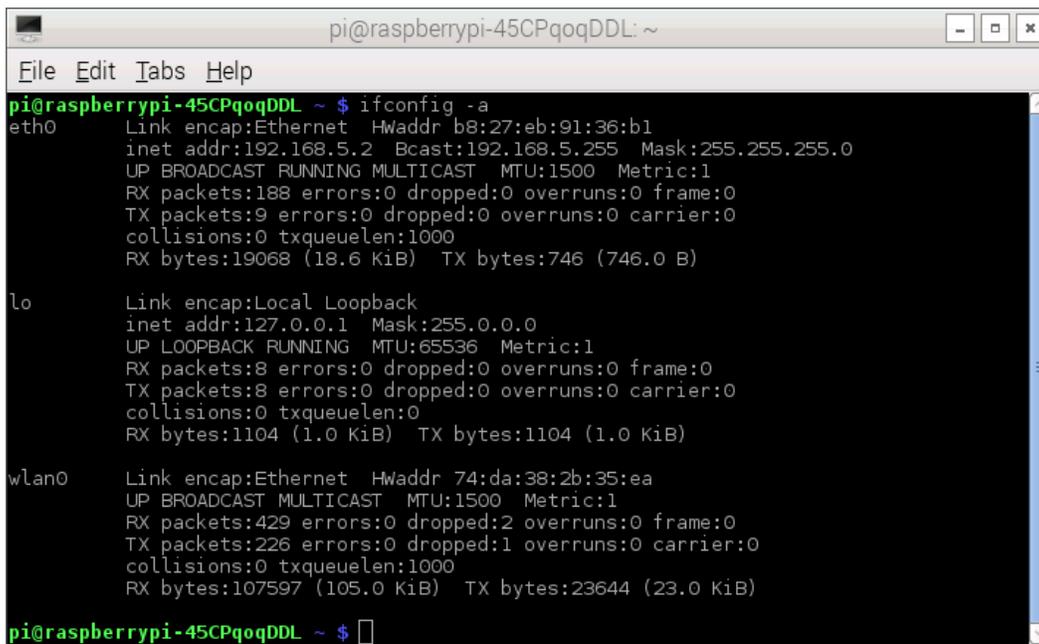
Paso 2. Utilizamos el siguiente comando para reiniciar la Raspberry Pi y que los cambios hechos hagan efecto.

```
sudo reboot
```

Verificar la Dirección IP en la Raspberry Pi

Una vez que hemos reiniciado la Raspberry debemos verificar si las modificaciones fueron correctas. Para esto debemos verificar la dirección IP que hemos asignado al dispositivo utilizando el siguiente comando:

```
ifconfig -a
```



```
pi@raspberrypi-45CPqoqDDL: ~  
File Edit Tabs Help  
pi@raspberrypi-45CPqoqDDL ~ $ ifconfig -a  
eth0      Link encap:Ethernet  Hwaddr b8:27:eb:91:36:b1  
          inet addr:192.168.5.2  Bcast:192.168.5.255  Mask:255.255.255.0  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:188 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:9 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:19068 (18.6 KiB)  TX bytes:746 (746.0 B)  
  
lo        Link encap:Local Loopback  
          inet addr:127.0.0.1  Mask:255.0.0.0  
          UP LOOPBACK RUNNING  MTU:65536  Metric:1  
          RX packets:8 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:0  
          RX bytes:1104 (1.0 KiB)  TX bytes:1104 (1.0 KiB)  
  
wlan0    Link encap:Ethernet  Hwaddr 74:da:38:2b:35:ea  
          UP BROADCAST MULTICAST  MTU:1500  Metric:1  
          RX packets:429 errors:0 dropped:2 overruns:0 frame:0  
          TX packets:226 errors:0 dropped:1 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:107597 (105.0 KiB)  TX bytes:23644 (23.0 KiB)  
  
pi@raspberrypi-45CPqoqDDL ~ $
```

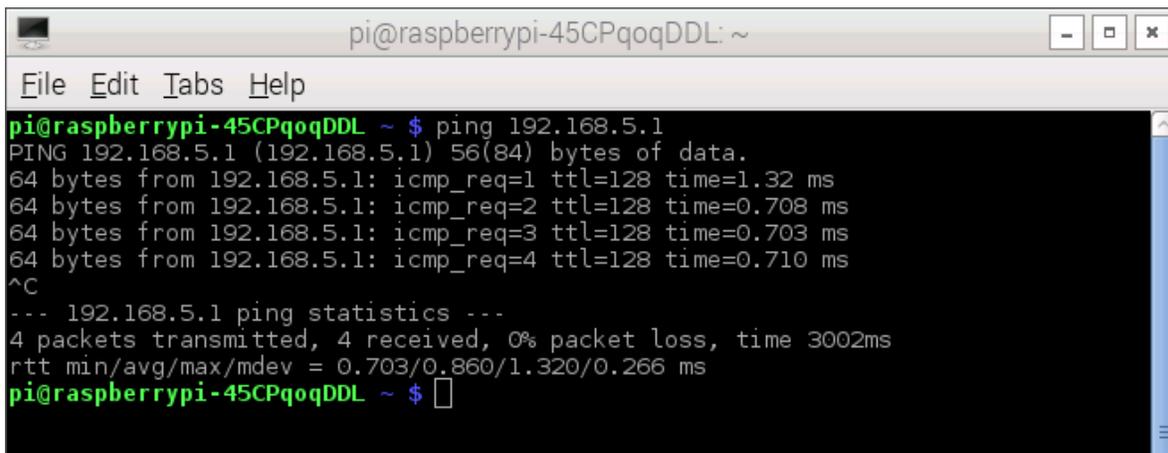
Si se encuentra conectado al PC, en la sección eth0 en la segunda línea debe contener la dirección IP de acuerdo al configurado, inet addr:192.168.5.1

Verificar la Dirección IP en la Raspberry Pi

Para verificar la configuración ha sido exitosa, lo único que debemos realizar será un ping desde la dirección IP del Raspberry Pi a la dirección del computador. En nuestro caso las direcciones asignadas al Raspberry Pi es 192.168.5.2 y la de nuestra PC con Windows es 192.168.5.2

```
ping 192.168.5.1
```

Después que nos aseguramos que hemos recibido 4 o 5 paquetes de datos de manera exitosa, presionamos Ctrl+C para cancelar, si todo ha funcionado correctamente, se espera tener como resultado lo siguiente:



```
pi@raspberrypi-45CPqoqDDL: ~  
File Edit Tabs Help  
pi@raspberrypi-45CPqoqDDL ~ $ ping 192.168.5.1  
PING 192.168.5.1 (192.168.5.1) 56(84) bytes of data.  
64 bytes from 192.168.5.1: icmp_req=1 ttl=128 time=1.32 ms  
64 bytes from 192.168.5.1: icmp_req=2 ttl=128 time=0.708 ms  
64 bytes from 192.168.5.1: icmp_req=3 ttl=128 time=0.703 ms  
64 bytes from 192.168.5.1: icmp_req=4 ttl=128 time=0.710 ms  
^C  
--- 192.168.5.1 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3002ms  
rtt min/avg/max/mdev = 0.703/0.860/1.320/0.266 ms  
pi@raspberrypi-45CPqoqDDL ~ $
```

Al realizar el comando ping desde la PC con Windows, es necesario ir a la línea de comando (CMD). Accedemos a ella presionando las teclas Windows+R y escriba CMD y luego Enter. Este proceso abrirá una ventana de línea de comandos. Ahora solo realizamos un ping hacia la dirección IP de la RaspPi, para nuestro caso será 192.168.5.2. Se espera tener el siguiente resultado:



```
Símbolo del sistema  
C:\Users\Lab-Digital>ping 192.168.5.2  
Haciendo ping a 192.168.5.2 con 32 bytes de datos:  
Respuesta desde 192.168.5.2: bytes=32 tiempo<1m TTL=64  
Estadísticas de ping para 192.168.5.2:  
Paquetes: enviados = 4, recibidos = 4, perdidos = 0  
(0% perdidos),  
Tiempos aproximados de ida y vuelta en milisegundos:  
Mínimo = 0ms, Máximo = 0ms, Media = 0ms  
C:\Users\Lab-Digital>
```

ANEXO C

Encuesta final aplicada a los estudiantes

Numero de encuesta: 1

Marque con una "X" la opción que crea conveniente.

Nivel de desempeño practico:

a) *¿Cuál es el nivel de satisfacción en la parte práctica que obtuvo con las guías de laboratorios?*

Muy insatisfecho Insatisfecho Satisfecho Muy Satisfecho

Nivel de aprendizaje:

1. *¿Cómo fue el aprendizaje obtenido en los laboratorios acerca del uso de módulo de cámara?*

Irrelevante Poco significativo Significativo Muy significativo

2. *¿Cómo fue el aprendizaje obtenido con el software MATLAB®?*

Irrelevante Poco significativo Significativo Muy significativo

Nivel de contenido teórico:

• *¿Cómo fue el contenido teórico de las guías de laboratorios?*

Muy insatisfecho Insatisfecho Satisfecho Muy Satisfecho

• *¿Qué tanto fue el aprendizaje que obtuvo en cada una de las prácticas de laboratorio?*

Laboratorio	Nada	Poco	Necesario	Suficiente
Nº 1: Captura y pre-procesamiento de las imágenes.			X	
Nº 2: Detección de los bordes de una imagen.		X		
No 3: Detección de Objetos a través de características.			X	
No 4: Detección de rostros y cuerpos.		X		

Limitaciones:

- ¿Cree que el tiempo fue suficiente para desarrollar satisfactoriamente el laboratorio?

Sí No

- ¿Fue útil el contenido teórico abordado en cada una de las guías de laboratorio?

Sí No

- ¿Es necesario utilizar más equipos para desarrollar las prácticas?

Sí No

Sugerencias:

¿Qué sugerencias recomienda para la mejora de las guías de laboratorio o para el desarrollo de las prácticas de laboratorio?

- **Mejorar la base teórica**
- **Si es posible más tiempo en los laboratorios**

Numero de encuesta: **2**

Marque con una "X" la opción que crea conveniente.

Nivel de desempeño practico:

b) *¿Cuál es el nivel de satisfacción en la parte práctica que obtuvo con las guías de laboratorios?*

Muy insatisfecho Insatisfecho Satisfecho **Muy Satisfecho**

Nivel de aprendizaje:

3. *¿Cómo fue el aprendizaje obtenido en los laboratorios acerca del uso de módulo de cámara?*

Irrelevante Poco significativo **Significativo** Muy significativo

4. *¿Cómo fue el aprendizaje obtenido con el software MATLAB®?*

Irrelevante Poco significativo Significativo **Muy significativo**

Nivel de contenido teórico:

• *¿Cómo fue el contenido teórico de las guías de laboratorios?*

Muy insatisfecho **Insatisfecho** Satisfecho Muy Satisfecho

• *¿Qué tanto fue el aprendizaje que obtuvo en cada una de las prácticas de laboratorio?*

Laboratorio	Nada	Poco	Necesario	Suficiente
Nº 1: Captura y pre-procesamiento de las imágenes.				X
Nº 2: Detección de los bordes de una imagen.		X		
No 3: Detección de Objetos a través de características.			X	
No 4: Detección de rostros y cuerpos.		X		

Limitaciones:

- ¿Cree que el tiempo fue suficiente para desarrollar satisfactoriamente el laboratorio?

Sí No

- ¿Fue útil el contenido teórico abordado en cada una de las guías de laboratorio?

Sí No

- ¿Es necesario utilizar más equipos para desarrollar las prácticas?

Sí No

Sugerencias:

¿Qué sugerencias recomienda para la mejora de las guías de laboratorio o para el desarrollo de las prácticas de laboratorio?

Numero de encuesta: **3**

Marque con una "X" la opción que crea conveniente.

Nivel de desempeño practico:

c) *¿Cuál es el nivel de satisfacción en la parte práctica que obtuvo con las guías de laboratorios?*

Muy insatisfecho Insatisfecho Satisfecho Muy Satisfecho

Nivel de aprendizaje:

5. *¿Cómo fue el aprendizaje obtenido en los laboratorios acerca del uso de módulo de cámara?*

Irrelevante Poco significativo Significativo Muy significativo

6. *¿Cómo fue el aprendizaje obtenido con el software MATLAB®?*

Irrelevante Poco significativo Significativo Muy significativo

Nivel de contenido teórico:

• *¿Cómo fue el contenido teórico de las guías de laboratorios?*

Muy insatisfecho Insatisfecho Satisfecho Muy Satisfecho

• *¿Qué tanto fue el aprendizaje que obtuvo en cada una de las prácticas de laboratorio?*

Laboratorio	Nada	Poco	Necesario	Suficiente
Nº 1: Captura y pre-procesamiento de las imágenes.		X		
Nº 2: Detección de los bordes de una imagen.		X		
No 3: Detección de Objetos a través de características.		X		
No 4: Detección de rostros y cuerpos.		X		

Limitaciones:

- ¿Cree que el tiempo fue suficiente para desarrollar satisfactoriamente el laboratorio?

Sí No

- ¿Fue útil el contenido teórico abordado en cada una de las guías de laboratorio?

Sí No

- ¿Es necesario utilizar más equipos para desarrollar las prácticas?

Sí No

Sugerencias:

¿Qué sugerencias recomienda para la mejora de las guías de laboratorio o para el desarrollo de las prácticas de laboratorio?

Más tiempo, más recursos y más laboratorios.

Numero de encuesta: 4

Marque con una "X" la opción que crea conveniente.

Nivel de desempeño practico:

d) *¿Cuál es el nivel de satisfacción en la parte práctica que obtuvo con las guías de laboratorios?*

Muy insatisfecho Insatisfecho Satisfecho Muy Satisfecho

Nivel de aprendizaje:

7. *¿Cómo fue el aprendizaje obtenido en los laboratorios acerca del uso de módulo de cámara?*

Irrelevante Poco significativo Significativo Muy significativo

8. *¿Cómo fue el aprendizaje obtenido con el software MATLAB®?*

Irrelevante Poco significativo Significativo Muy significativo

Nivel de contenido teórico:

• *¿Cómo fue el contenido teórico de las guías de laboratorios?*

Muy insatisfecho Insatisfecho Satisfecho Muy Satisfecho

• *¿Qué tanto fue el aprendizaje que obtuvo en cada una de las prácticas de laboratorio?*

Laboratorio	Nada	Poco	Necesario	Suficiente
Nº 1: Captura y pre-procesamiento de las imágenes.		X		
Nº 2: Detección de los bordes de una imagen.		X		
No 3: Detección de Objetos a través de características.		X		
No 4: Detección de rostros y cuerpos.		X		

Limitaciones:

- ¿Cree que el tiempo fue suficiente para desarrollar satisfactoriamente el laboratorio?

Sí No

- ¿Fue útil el contenido teórico abordado en cada una de las guías de laboratorio?

Sí No

- ¿Es necesario utilizar más equipos para desarrollar las prácticas?

Sí No

Sugerencias:

¿Qué sugerencias recomienda para la mejora de las guías de laboratorio o para el desarrollo de las prácticas de laboratorio?

Dentro de las recomendaciones diría que un poco más de tiempo con mayores revisiones del tutor.

Numero de encuesta: **5**

Marque con una "X" la opción que crea conveniente.

Nivel de desempeño practico:

e) *¿Cuál es el nivel de satisfacción en la parte práctica que obtuvo con las guías de laboratorios?*

Muy insatisfecho Insatisfecho Satisfecho Muy Satisfecho

Nivel de aprendizaje:

9. *¿Cómo fue el aprendizaje obtenido en los laboratorios acerca del uso de módulo de cámara?*

Irrelevante Poco significativo Significativo Muy significativo

10. *¿Cómo fue el aprendizaje obtenido con el software MATLAB®?*

Irrelevante Poco significativo Significativo Muy significativo

Nivel de contenido teórico:

• *¿Cómo fue el contenido teórico de las guías de laboratorios?*

Muy insatisfecho Insatisfecho Satisfecho Muy Satisfecho

• *¿Qué tanto fue el aprendizaje que obtuvo en cada una de las prácticas de laboratorio?*

Laboratorio	Nada	Poco	Necesario	Suficiente
Nº 1: Captura y pre-procesamiento de las imágenes.		X		
Nº 2: Detección de los bordes de una imagen.		X		
No 3: Detección de Objetos a través de características.		X		
No 4: Detección de rostros y cuerpos.		X		

Limitaciones:

- ¿Cree que el tiempo fue suficiente para desarrollar satisfactoriamente el laboratorio?

Sí No

- ¿Fue útil el contenido teórico abordado en cada una de las guías de laboratorio?

Sí No

- ¿Es necesario utilizar más equipos para desarrollar las prácticas?

Sí No

Sugerencias:

¿Qué sugerencias recomienda para la mejora de las guías de laboratorio o para el desarrollo de las prácticas de laboratorio?

Más graficas e imágenes que sean explicitas sobre los temas a abordar en las prácticas de laboratorio.

Numero de encuesta: **6**

Marque con una "X" la opción que crea conveniente.

Nivel de desempeño practico:

f) *¿Cuál es el nivel de satisfacción en la parte práctica que obtuvo con las guías de laboratorios?*

Muy insatisfecho Insatisfecho Satisfecho Muy Satisfecho

Nivel de aprendizaje:

11. *¿Cómo fue el aprendizaje obtenido en los laboratorios acerca del uso de módulo de cámara?*

Irrelevante Poco significativo Significativo Muy significativo

12. *¿Cómo fue el aprendizaje obtenido con el software MATLAB®?*

Irrelevante Poco significativo Significativo Muy significativo

Nivel de contenido teórico:

• *¿Cómo fue el contenido teórico de las guías de laboratorios?*

Muy insatisfecho Insatisfecho Satisfecho Muy Satisfecho

• *¿Qué tanto fue el aprendizaje que obtuvo en cada una de las prácticas de laboratorio?*

Laboratorio	Nada	Poco	Necesario	Suficiente
Nº 1: Captura y pre-procesamiento de las imágenes.			X	
Nº 2: Detección de los bordes de una imagen.				X
No 3: Detección de Objetos a través de características.			X	
No 4: Detección de rostros y cuerpos.				X

Limitaciones:

- ¿Cree que el tiempo fue suficiente para desarrollar satisfactoriamente el laboratorio?

Sí No

- ¿Fue útil el contenido teórico abordado en cada una de las guías de laboratorio?

Sí No

- ¿Es necesario utilizar más equipos para desarrollar las prácticas?

Sí No

Sugerencias:

¿Qué sugerencias recomienda para la mejora de las guías de laboratorio o para el desarrollo de las prácticas de laboratorio?

Numero de encuesta: 7

Marque con una "X" la opción que crea conveniente.

Nivel de desempeño practico:

g) *¿Cuál es el nivel de satisfacción en la parte práctica que obtuvo con las guías de laboratorios?*

Muy insatisfecho Insatisfecho Satisfecho Muy Satisfecho

Nivel de aprendizaje:

13. *¿Cómo fue el aprendizaje obtenido en los laboratorios acerca del uso de módulo de cámara?*

Irrelevante Poco significativo Significativo Muy significativo

14. *¿Cómo fue el aprendizaje obtenido con el software MATLAB®?*

Irrelevante Poco significativo Significativo Muy significativo

Nivel de contenido teórico:

• *¿Cómo fue el contenido teórico de las guías de laboratorios?*

Muy insatisfecho Insatisfecho Satisfecho Muy Satisfecho

• *¿Qué tanto fue el aprendizaje que obtuvo en cada una de las prácticas de laboratorio?*

Laboratorio	Nada	Poco	Necesario	Suficiente
Nº 1: Captura y pre-procesamiento de las imágenes.			X	
Nº 2: Detección de los bordes de una imagen.				X
No 3: Detección de Objetos a través de características.			X	
No 4: Detección de rostros y cuerpos.				X

Limitaciones:

- ¿Cree que el tiempo fue suficiente para desarrollar satisfactoriamente el laboratorio?

Sí No

- ¿Fue útil el contenido teórico abordado en cada una de las guías de laboratorio?

Sí No

- ¿Es necesario utilizar más equipos para desarrollar las prácticas?

Sí No

Sugerencias:

¿Qué sugerencias recomienda para la mejora de las guías de laboratorio o para el desarrollo de las prácticas de laboratorio?

Aumentar el número de laboratorios y la parte teórica.

Numero de encuesta: **8**

Marque con una "X" la opción que crea conveniente.

Nivel de desempeño practico:

h) ¿Cuál es el nivel de satisfacción en la parte práctica que obtuvo con las guías de laboratorios?

Muy insatisfecho Insatisfecho Satisfecho Muy Satisfecho

Nivel de aprendizaje:

15. ¿Cómo fue el aprendizaje obtenido en los laboratorios acerca del uso de módulo de cámara?

Irrelevante Poco significativo Significativo Muy significativo

16. ¿Cómo fue el aprendizaje obtenido con el software MATLAB®?

Irrelevante Poco significativo Significativo Muy significativo

Nivel de contenido teórico:

• *¿Cómo fue el contenido teórico de las guías de laboratorios?*

Muy insatisfecho Insatisfecho Satisfecho Muy Satisfecho

• *¿Qué tanto fue el aprendizaje que obtuvo en cada una de las prácticas de laboratorio?*

Laboratorio	Nada	Poco	Necesario	Suficiente
Nº 1: Captura y pre-procesamiento de las imágenes.				X
Nº 2: Detección de los bordes de una imagen.				X
No 3: Detección de Objetos a través de características.			X	
No 4: Detección de rostros y cuerpos.			X	

Limitaciones:

- ¿Cree que el tiempo fue suficiente para desarrollar satisfactoriamente el laboratorio?

Sí No

- ¿Fue útil el contenido teórico abordado en cada una de las guías de laboratorio?

Sí No

- ¿Es necesario utilizar más equipos para desarrollar las prácticas?

Sí No

Sugerencias:

¿Qué sugerencias recomienda para la mejora de las guías de laboratorio o para el desarrollo de las prácticas de laboratorio?

Mejores y más equipos.

Numero de encuesta: **9**

Marque con una "X" la opción que crea conveniente.

Nivel de desempeño practico:

i) *¿Cuál es el nivel de satisfacción en la parte práctica que obtuvo con las guías de laboratorios?*

Muy insatisfecho Insatisfecho Satisfecho Muy Satisfecho

Nivel de aprendizaje:

17. *¿Cómo fue el aprendizaje obtenido en los laboratorios acerca del uso de módulo de cámara?*

Irrelevante Poco significativo Significativo Muy significativo

18. *¿Cómo fue el aprendizaje obtenido con el software MATLAB®?*

Irrelevante Poco significativo Significativo Muy significativo

Nivel de contenido teórico:

• *¿Cómo fue el contenido teórico de las guías de laboratorios?*

Muy insatisfecho Insatisfecho Satisfecho Muy Satisfecho

• *¿Qué tanto fue el aprendizaje que obtuvo en cada una de las prácticas de laboratorio?*

Laboratorio	Nada	Poco	Necesario	Suficiente
Nº 1: Captura y pre-procesamiento de las imágenes.			X	
Nº 2: Detección de los bordes de una imagen.				X
No 3: Detección de Objetos a través de características.			X	
No 4: Detección de rostros y cuerpos.			X	

Limitaciones:

- ¿Cree que el tiempo fue suficiente para desarrollar satisfactoriamente el laboratorio?

Sí No

- ¿Fue útil el contenido teórico abordado en cada una de las guías de laboratorio?

Sí No

- ¿Es necesario utilizar más equipos para desarrollar las prácticas?

Sí No

Sugerencias:

¿Qué sugerencias recomienda para la mejora de las guías de laboratorio o para el desarrollo de las prácticas de laboratorio?

Ir aumentando poco a poco la complejidad y el reto que estas guías representan.

Numero de encuesta: **10**

Marque con una "X" la opción que crea conveniente.

Nivel de desempeño practico:

j) *¿Cuál es el nivel de satisfacción en la parte práctica que obtuvo con las guías de laboratorios?*

Muy insatisfecho Insatisfecho Satisfecho Muy Satisfecho

Nivel de aprendizaje:

19. *¿Cómo fue el aprendizaje obtenido en los laboratorios acerca del uso de módulo de cámara?*

Irrelevante Poco significativo Significativo Muy significativo

20. *¿Cómo fue el aprendizaje obtenido con el software MATLAB®?*

Irrelevante Poco significativo Significativo Muy significativo

Nivel de contenido teórico:

• *¿Cómo fue el contenido teórico de las guías de laboratorios?*

Muy insatisfecho Insatisfecho Satisfecho Muy Satisfecho

• *¿Qué tanto fue el aprendizaje que obtuvo en cada una de las prácticas de laboratorio?*

Laboratorio	Nada	Poco	Necesario	Suficiente
Nº 1: Captura y pre-procesamiento de las imágenes.			X	
Nº 2: Detección de los bordes de una imagen.				X
No 3: Detección de Objetos a través de características.			X	
No 4: Detección de rostros y cuerpos.			X	

Limitaciones:

- ¿Cree que el tiempo fue suficiente para desarrollar satisfactoriamente el laboratorio?

Sí No

- ¿Fue útil el contenido teórico abordado en cada una de las guías de laboratorio?

Sí No

- ¿Es necesario utilizar más equipos para desarrollar las prácticas?

Sí No

Sugerencias:

¿Qué sugerencias recomienda para la mejora de las guías de laboratorio o para el desarrollo de las prácticas de laboratorio?