



UNIVERSIDAD DE JAÉN
Nombre del Centro

Trabajo Fin de Grado

SISTEMA AUTÓNOMO DE DETECCIÓN DE PLAGAS EN EL OLIVAR

Alumno: Daniel Fernando Sánchez de la Cuadra

Tutor: Prof. D. Silvia María Satorres Martínez
Prof. D. Pablo Cano Marchal

Dpto: Departamento de Ingeniería Electrónica y
Automática



Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Ingeniería Electrónica y Automática

Doña Silvia María Satorres Martínez y Don Pablo Cano Marchal, tutores del Trabajo Fin de Grado titulado: SISTEMA AUTÓNOMO DE DETECCIÓN DE PLAGAS EN EL OLIVAR, que presenta Daniel Fernando Sánchez de la Cuadra, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, 22 junio de 2022

El alumno:

Daniel Fernando Sánchez de la Cuadra

Los tutores:

**SATORRES
MARTINEZ
SILVIA -
26741651B**

Firmado digitalmente por
SATORRES MARTINEZ SILVIA -
26741651B
Nombre de reconocimiento
(DN): c=ES,
serialNumber=IDCES-2674165
1B, givenName=SILVIA,
sn=SATORRES MARTINEZ,
cn=SATORRES MARTINEZ
SILVIA - 26741651B
Fecha: 2022.06.22 11:05:15
+02'00'

Silvia María Satorres Martínez

**CANO MARCHAL
PABLO -
77344429Y**

Firmado digitalmente
por CANO MARCHAL
PABLO - 77344429Y
Fecha: 2022.06.22
10:39:20 +02'00'

Pablo Cano Marchal

Índice

1. INTRODUCCIÓN	1
1.1. Motivación	1
1.2. Objetivos	1
1.3. Metodología a desarrollar	2
1.4. Estructura	3
2. Plagas en el olivar y su detección	4
2.1. Tipos de plagas	4
2.2. Mosca del olivo (<i>Bactrocera Oleae</i>)	7
2.2.1. Ciclo biológico	8
2.2.2. Desarrollo de la plaga	9
2.2.3. Daños	10
2.2.4. Seguimiento de la plaga	11
2.2.5. Medios de lucha	13
2.3. Sistemas de detección automáticos	14
3. Hardware del prototipo	16
3.1. Controlador	16
3.1.1. Requerimientos	16
3.1.2. Optimización del consumo	18
3.2. Conexiones red	19
3.3. Sistema de Visión	21
3.3.1. Requerimientos climáticos	21
3.3.2. Requerimientos de visión por computador	23
3.3.3. Elección de cámara	25
3.4. Fuente de alimentación	27
3.4.1. Consumo del sistema	27
3.4.2. Dimensionamiento de la placa fotovoltaica	28
3.4.3. Dimensionamiento de la batería	29
3.4.4. Regulador de carga	30
3.5. Montaje del sistema	31
4. Software del prototipo	33
4.1. Programación Raspberry	33
4.1.1. Instalación y puesta a punto del sistema operativo	33
4.1.2. Conexión y configuración de la cámara	34
4.1.3. Rutina de adquisición y envío de imágenes	36
4.2. Detector de plagas	41

4.2.1.	Obtención de imágenes	41
4.2.2.	Clasificación mediante redes neuronales	44
5.	Resultados	53
6.	Presupuesto	55
7.	Conclusiones y trabajos futuros	56
7.1.	Posible diseño futuro	56
	Bibliografía	57

Índice de figuras

Figura 1.1. Distintas plagas de insectos en el olivar	5
Figura 1.2. Plagas de Hongos en el olivar	5
Figura 1.3. Xylella fastidiosa	6
Figura 2.4. Hábitat Bactrocera Oleae	7
Figura 2.5. Bactrocera Oleae	7
Figura 2.6. Huevo de la mosca	8
Figura 2.7. Pupario	8
Figura 2.8. Adulto saliendo del pupario	8
Figura 2.9. Dinámica poblacional	9
Figura 2.10. Olivo dañado por Bactrocera Oleae	10
Figura 2.11. Trampa cromotrópica	11
Figura 2.12. NANOENVI AG	14
Figura 2.13. Cámara trampa SEMIOS	14
Figura 2.14. Trampa inteligente	15
Figura 3.15. Consumo en funcionamiento	17
Figura 3.16. Compresión de archivos	17
Figura 3.17. Velocidad de conexión Ethernet	17
Figura 3.18. Temperatura en funcionamiento	18
Figura 3.19. Raspberry Pi 3 B+	18
Figura 3.20. PiWatcher	19
Figura 3.21. Conexiones red	20
Figura 3.22. Temperaturas anuales en Jaén	21
Figura 3.23. Precipitaciones anuales en Jaén	21
Figura 3.24. Diseño caja protectora 3D	22
Figura 3.25. Salida y puesta del sol en Jaén	22
Figura 3.26. Nubosidad anual en Jaén	22
Figura 3.27. Trampa cromotrópica usada	23
Figura 3.28. Campo de visión con 1 cámara	23
Figura 3.29. Campo de visión con 2 cámara	24
Figura 3.30. Mosca del olivo	24
Figura 3.31. Modelo Pin-Hole	25
Figura 3.32. Conexiones Raspberry	25
Figura 3.33. Cámara ELP 8MP	26
Figura 3.34. Radiación por la Agencia Andaluza y por el atlas de la radiación.	28
Figura 3.35. Módulo fotovoltaico de 15W.	29
Figura 3.36. Batería 12V y 18Ah	30
Figura 3.37. Regulador de carga	30
Figura 3.38. Estructura SkechUp	31
Figura 3.39. Estructura	32
Figura 4.40. Versión básica de Raspbian	33
Figura 4.41. BalenaEtcher	34
Figura 4.42. Prueba de captura de imagen	35
Figura 4.43. Mensaje en la bandeja de entrada Gmail	38
Figura 4.44. Crontab	39
Figura 4.45. Configuración del momento de ejecución Crontab	39
Figura 4.46. Crontab guru	40

Figura 4.47. Insectos capturados	41
Figura 4.48. Adquisición de imágenes	42
Figura 4.49. Distintos insectos capturados	43
Figura 4.50. Distorsión de barril	44
Figura 4.51. Camera Calibrator	44
Figura 4.52. Imagen segmentada manualmente	45
Figura 4.53. Arquitectura de una Red neuronal	46
Figura 4.54. Guide Entrenamiento	47
Figura 4.55. Matriz de confusión de la geometría	49
Figura 4.56. Posibles candidatos	49
Figura 4.57. Matriz de confusión de color	51
Figura 4.58. Resultado del Análisis	52
Figura 4.59. Sistema autónomo de detección de plagas	53

Índice de Ecuaciones

Ecuación 2.1. Mosca por Trampa y Día	12
Ecuación 3.2. triángulos semejantes	25
Ecuación 3.3. Cálculo del consumo diario	27
Ecuación 3.4. Potencia a instalar	28
Ecuación 3.5. Capacidad de la batería	29

Índice de Tablas

Tabla 1. Ejemplo de capturas	12
Tabla 4.2. Test de detección de moscas	54

Índice de Diagramas de flujo

Diagrama 4.1. Rutina.py	36
Diagrama 4.2. Entrenamiento	48
Diagrama 4.3. Análisis de la imagen	50

Agradecimientos

Me gustaría agradecer toda la ayuda recibida a lo largo del Grado, tanto a mis profesores como a mis compañeros y familia.

En primer lugar, agradecer a mis tutores Silvia María Satorres Martínez y Pablo Cano Marcha por guiarme en la resolución y redacción de este TFG, enseñándome tantas cosas durante el proyecto.

Agradecer a Diego Manuel Martínez Gila por todos sus consejos y apoyo mientras desarrollaba el Hardware, así como a Jorge Aguilera Tejero por sus consejos sobre de la alimentación fotovoltaica, a Alejandro Sánchez García por permitirme reutilizar materiales de otros proyectos y a Gustavo Medina Sánchez por facilitarme el acceso al laboratorio de mecánica para mecanizar la estructura.

Agradezco también el apoyo y las ideas que me daban mis compañeros Pablo, Antonio y Jesús, sus puntos de vista me permitían reflexionar mis ideas y darles una vuelta para mejorar los resultados obtenidos.

Y por último agradecer todo el amor y cariño recibido por mi familia en este año, sin duda su apoyo ha sido crucial.

1. INTRODUCCIÓN

1.1. Motivación

La motivación de este trabajo está dividida en varias partes. En primera instancia mencionar que este es mi Trabajo Fin de Grado, en el cual expreso parte de todo lo que he aprendido a lo largo del Grado de Ingeniería electrónica, y sirve para poner fin a esta etapa de mi formación, aventurándome en la siguiente.

Durante la búsqueda del que iba a ser mi TFG hablé con varios profesores buscando una propuesta interesante y desafiante. Fue así como conocí la propuesta del grupo de robótica, automática y visión por computador, la cual consistía en un sistema de detección de plagas para el olivar.

Esta propuesta me gustó porque recoge varios de los valores que tengo como persona, como son el amor por mi tierra, Jaén, el amor por la naturaleza, la optimización de recursos y la pasión por el campo, que me devuelve a mis raíces y seres queridos.

Por último, mencionar todos los objetivos de desarrollo sostenible que aportan este trabajo, como son:

- Energía sostenible y no contaminante.
- Producción y consumo responsables.
- Vida de ecosistemas terrestres.

Estos objetivos son fundamentales a la hora de realizar proyectos de forma responsable, fomentando el desarrollo sostenible.

1.2. Objetivos

El objetivo de este trabajo es diseñar un sistema autónomo capaz de detectar el aumento de las poblaciones de *Bactrocera Oleae* en las estaciones de control biológico sin la necesidad de un operario en la zona de estudio, consiguiendo así una mayor eficacia en la prevención de plagas y una disminución del número de viajes a la zona de estudio, reduciendo la huella de carbono del transporte y facilitando el trabajo de los biólogos y agricultores.

Este sistema realiza la adquisición de imagen de las trampas cromotrópicas, envía por correo las imágenes y son posteriormente analizadas con un software para determinar si el MDT (Moscas por Día y Trampa) está aumentando.

Entendiendo lo anterior, los objetivos específicos del trabajo son:

- Dimensionar y diseñar un prototipo autónomo para la detección de las plagas en el olivar.
- Programar el prototipo para enviar imágenes periódicas de las trampas por correo.
- Programar un software capaz de analizar la cantidad de insectos de las imágenes mediante técnicas de visión por computador.
- Validar el sistema al completo en el campo.

1.3. Metodología a desarrollar

Con el fin de cumplir los objetivos anteriormente mencionados se desarrollará la siguiente metodología:

- Dimensionar y buscar todos los componentes del hardware para un correcto funcionamiento conjunto. Estos elementos se situarán dentro de una carcasa protectora y sobre un soporte diseñados en 3D para su posterior colocación en el campo.
- Realizar y comprobar el correcto funcionamiento de las conexiones entre los distintos elementos del Hardware. Se comprobarían tanto las conexiones de la cámara y el temporizador con la Raspberry como las conexiones de la fuente de alimentación.
- Diseñar el software del controlador para realizar la tarea de adquirir la imagen y enviarla por correo. Para esto se programará una tarea en Linux para adquirir la imagen y se enviará por correo a través del protocolo SMTP.
- Diseñar un software de detección de *Bactrocera Oleae* en Matlab utilizando técnicas de visión por computador. Este software tendrá 2 modos: clasificador paramétrico y redes neuronales.

1.4. Estructura

La memoria consta de 7 capítulos estructurados de la siguiente manera:

- Capítulo 1. Introducción: Se explican los motivos de este trabajo, el problema que se intenta resolver y como se va a afrontar.
- Capítulo 2. Plagas en el olivar y su detección: Se explican las plagas más comunes del olivar, haciendo hincapié en la plaga de *Bactrocera Oleae*, mostrando los sistemas utilizados para su control.
- Capítulo 3. Hardware del prototipo: Desglose de todos los materiales necesarios para construir el hardware, justificando cada elección.
- Capítulo 4. Software del prototipo: Tanto la programación del Hardware como el software de análisis posterior.
- Capítulo 5. Resultados: Porcentaje de aciertos.
- Capítulo 6. Presupuesto: Desglose de precios de todos los materiales utilizados para el desarrollo del prototipo y costes indirectos.
- Capítulo 7. Conclusiones y trabajos futuros: Métodos de análisis alternativos y posibles mejoras del sistema.

2. Plagas en el olivar y su detección

La provincia de Jaén posee el olivar más extenso del mundo, un ecosistema con una gran biodiversidad en el que conviven varias especies de animales e insectos, en el que se producen más de 400.000 toneladas de aceite de oliva al año, lo cual sitúa Jaén como la principal exportadora a nivel mundial.

Cuidar este extenso olivar requiere un gran esfuerzo por parte de los agricultores de nuestra tierra. Al regadío y la poda hay que sumarle los distintos tratamientos fitosanitarios que hay que realizar para garantizar la calidad de la aceituna. Este TFG se centrará en la detección de plagas, en concreto la plaga de la mosca del olivo, la cual causa graves daños tanto a la cantidad como a la calidad del aceite de oliva.

2.1. Tipos de plagas

· Plagas de insectos: (Fig. 1.1)

-Mosca del olivo (*Bactrocera Oleae*) (Fig. 1.1, 1): Esta mosca deposita sus huevos dentro de las aceitunas, de modo que su larva se alimenta de la aceituna, reduciendo la masa de la misma y reduciendo la calidad del aceite recolectado. Esta plaga se explicará con más detalle en el punto 2.2.

-Polilla del olivo (*Prays oleae*) (Fig. 1.1, 2): Es la plaga más común en La Rioja y se alimenta de las hojas, de los botones florales e incluso del fruto, por lo que pueden llegar a ocasionar severas pérdidas económicas. El tratamiento puede realizarse con diversos insecticidas como Delfin, Mospilan Max...

-Abichado del olivo (*Euzophera pinguis*) (Fig. 1.1, 3): En estado larval se alimenta de la corteza del olivo, dañando los flujos de savia. En los peores casos esto desemboca en la debilitación de la zona afectada secándose. Suele tratarse con depredación natural por parte de otras especies como *Iconella Myelolenta*, pero lo más importante es tratar las heridas de los olivos para evitar la puesta del huevo en zonas profundas.

-Barrenillo del olivo (*Phloeotribus scarabaeoides*) (Fig. 1.1, 4): Es una plaga muy común de baja incidencia. Las larvas se alimentan de la madera de las ramas partidas ó podadas y generan una serie de galerías que, en el caso de infectar a un árbol, debilitan al olivo pudiendo provocarle la muerte. Se previene con la poda, ya que el adulto prefiere depositar sus huevos en ramas que estén secándose.

-Cochinilla de la tizne (*Saissetia oleae*) (Fig. 1.1, 5): Se alimenta succionando la savia, de forma que debilita al olivo cuando el ataque es severo. Puede tratarse con insecticidas y con la poda racional, ya que el sol puede reducir severamente la población de cochinillas.

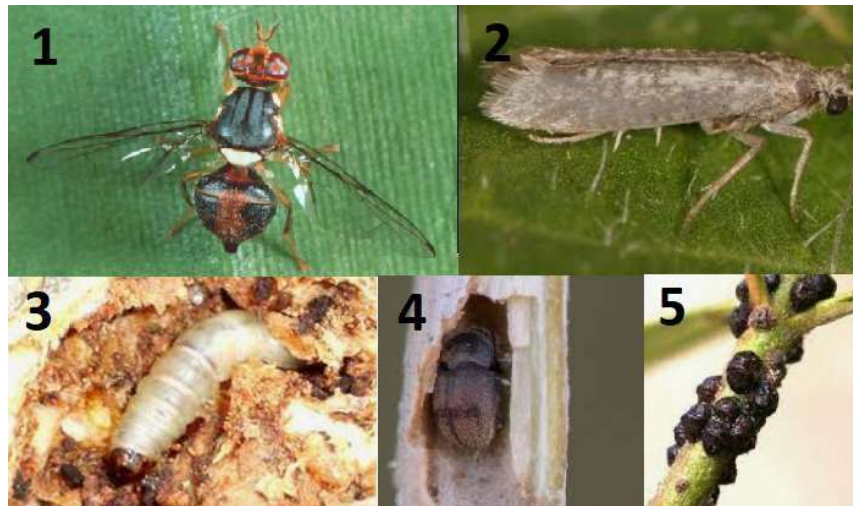


Figura 1.1. Distintas plagas de insectos en el olivar

· Plagas de hongos:

- repilo del olivo (*Fusicladium oleagineum*) (Fig. 1.2, 1): Es una enfermedad causada por hongos que deja unas marcas circulares en las hojas del olivo y aparece cuando las temperaturas son suaves y hay humedad. Se trata con fungicidas.

-Verticilosis (*Verticillium dahliae*) (Fig. 1.2, 2): Es una enfermedad causada por un hongo. Afecta al sistema vascular de la planta y la seca, llegando incluso a matarla. Los olivos utilizan una estrategia natural por la cual combaten el hongo gracias a los microorganismos que viven en el suelo, por lo que promover la riqueza biológica del suelo es la clave contra este tipo de enfermedades [2].



Figura 1.2. Plagas de Hongos en el olivar

· Plagas emergentes:

- *Xylella fastidiosa* (Fig. 1.3.): Es una bacteria que se encuentra en la xilema¹ y puede llegar a taponar los vasos, obstruyendo el flujo de savia. Esta plaga apareció en 2019 y se ha ido extendiendo por Europa y actualmente afecta a Baleares. Si se detecta en un olivo, es erradicado junto con los que se encuentren en un radio de 50m del mismo, además de establecer un perímetro de estudio de 2,5 Km de la zona estudiada [7].



Figura 1.3. *Xylella fastidiosa*

¹ Tejido conductor encargado de transportar nutrientes desde la raíz a todas las partes de la planta.

2.2. Mosca del olivo (*Bactrocera Oleae*)

Esta mosca pertenece a la familia de los Tripéptidos y vive principalmente en los alrededores del mar mediterráneo, aunque existen poblaciones en Asia occidental y Centroamérica (Fig. 2.4.).

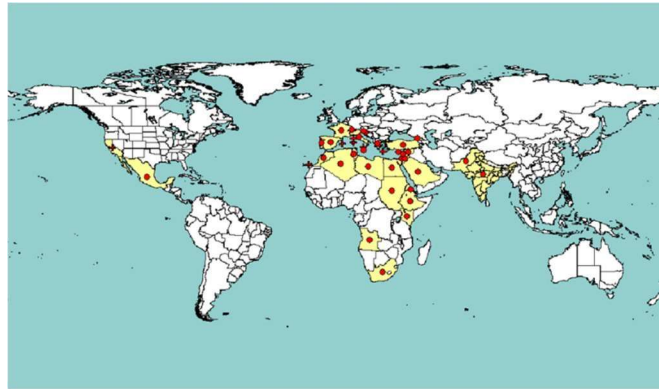


Figura 2.4. Hábitat *Bactrocera Oleae*

Se trata de una mosca que en edad adulta mide entre 4 y 5 milímetros de longitud (Fig. 2.5.). Tienen un cuerpo con varios tonos marrones y destaca una mancha negra en la punta de sus alas y un triángulo amarillento sobre el dorso que se extiende a la base de las alas. La hembra es algo mayor que el macho y tiene al final del abdomen un ovipositor² muy visible [1].



Figura 2.5. *Bactrocera Oleae*

² Órgano reproductor de las hembras que permite depositar huevos.

2.2.1. Ciclo biológico

Tras el apareamiento con el macho, la hembra elige un fruto de buen tamaño (entre 8 y 10mm), que no haya sido picado por otra hembra y comienza la puesta. Perfora la piel de la oliva con su ovipositor y coloca 1 único huevo de color blanco lechoso y de longitud inferior a 1mm (Fig. 2.6.).



Figura 2.6. Huevo de la mosca

Cuando el huevo madura nace la larva, la cual comienza a alimentarse de la aceituna desde su interior. Cuando ha comido suficiente alcanza un tamaño de unos 6-8mm y procede con el paso al estado pupal. Se acerca a la superficie de la aceituna y forma el pupario (Fig. 2.7.), donde comenzará la metamorfosis.



Figura 2.7. Pupario

Una vez completada la metamorfosis, el adulto rompe el pupario (Fig. 2.8.) con un movimiento brusco y sale al exterior. En ocasiones, una mala colocación del pupario impide a la mosca salir y queda atrapada dentro de la aceituna.



Figura 2.8. Adulto saliendo del pupario

Una vez que el adulto está fuera necesita alimentarse de sustancias azucaradas para desarrollar sus genitales y producir las feromonas con las que atraerse. Tras la cópula la hembra tarda de 2 a 3 días en poner el huevo en una aceituna nueva [6].

2.2.2. Desarrollo de la plaga

Durante un año se producen de 3 a 5 generaciones, las cuales varían de tamaño según diversos factores ambientales, aun que destacan 2 picos (Fig. 2.9.): uno en primavera que se corresponde a la generación invernante y la segunda, más intensa, a principios de otoño, coincidiendo con la bajada de temperaturas y un aumento de la humedad.

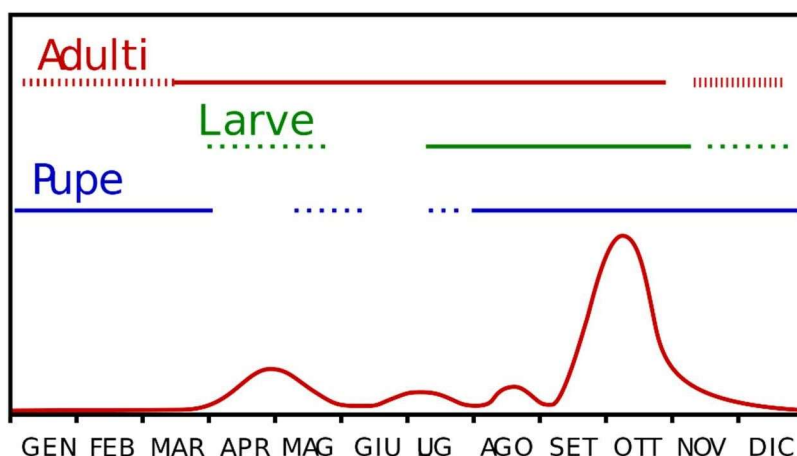


Figura 2.9. Dinámica poblacional

Y es que los principales factores que afectan a esta plaga son meteorológicos, ya que las condiciones críticas ideales para el desarrollo de la plaga son temperaturas moderadas (por debajo de los 32-34°C) y clima húmedo, por ello no es de extrañar que se encuentren más presentes en el litoral que en el centro de España.

Influye también la variedad de olivo (atacando especialmente variedades de olivo precoces y para mesa) y la relación con alternancia o vecería en la producción, fenómeno muy propenso en cultivos de secano en los cuales, los años de escasa producción son precedidos por un año de alta producción, del cual suele quedar bastante aceituna sin recolectar.

En estas aceitunas, ponen sus huevos las moscas y tras el invierno aumenta la población de las mismas. Debido a esto caen más aceitunas y por lo tanto quedan

menos para que pongan huevos ese año, por lo que el año siguiente la plaga será menor [1] [3].

2.2.3. Daños

La *Bactrocera Oleae* es una de las plagas más importantes y temibles del olivo debido a la necesidad de un fruto como la aceituna para la puesta del huevo y el crecimiento de la larva, las aceitunas picadas (Fig. 2.10.) presentan en torno a un 20% menos de masa, lo cual reduce el rendimiento de la plantación y la calidad del aceite notablemente.

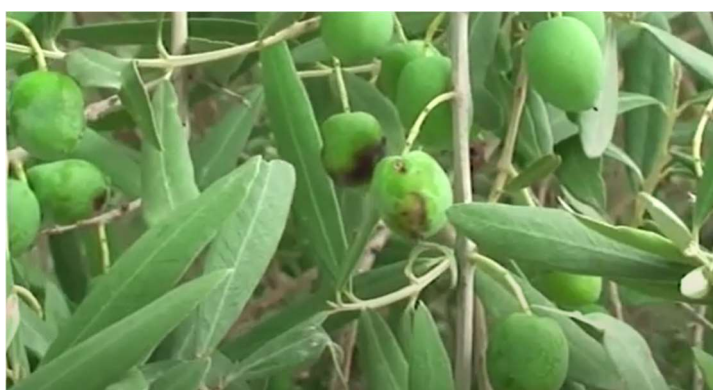


Figura 2.10. Olivo dañado por *Bactrocera Oleae*

El daño más directo es la disminución del peso recolectado, ya que en muchos casos los daños de la larva ocasionan la caída del fruto al suelo y por lo tanto el fruto es completamente desechado.

Por otro lado, la calidad de la aceituna recolectada disminuye considerablemente, ya que las galerías creadas por la larva permiten a hongos y bacterias desarrollarse dentro de la aceituna, aumentando la acidez y deteriorando las características organolépticas de la misma.

En el caso del olivar de almazara se considera que a partir de un 10% de frutos atacados por la mosca ya hay pérdidas económicas, mientras que en el olivar de verdeo cada fruto picado es totalmente desechado y no sirve para su aderezo [2] [3].

2.2.4. Seguimiento de la plaga

Desde 2008 es obligatorio por ley realizar seguimientos de los niveles poblacionales para estimar el riesgo de plagas y enfermedades en el olivo y facilitar los resultados a la RAIF (Red de alerta en información fitosanitaria).

Para una realización correcta de la estrategia de control es necesario realizar los siguientes pasos:

- Selección de la estación de control biológico (ECB): Debe ser representativa respecto a la parcela (edad y variante de la planta, tipo de suelo, clima, laboreo...). En el caso del olivo debe colocarse por ley 1 estación por cada 300 ha.

- Instalación de trampas: En el caso de *Bactrocera Oleae* se utiliza la trampa cromotrópica (Fig. 2.11.), la cual consiste en una placa engomada amarilla cubierta de feromonas, las cuales atraen principalmente adultos machos.



Figura 2.11. Trampa cromotrópica

Se instalan orientadas al suroeste del exterior del árbol durante toda la campaña, procurando limpiar la periferia para evitar que se peguen hojas. La feromona se sitúa encima sujeta por un alambre y cada estación debe separarse por lo menos 40m.

Las trampas se instalan después del cuajado y antes del endurecimiento del hueso, en junio y se recogen 15 días antes de la recolección.

Tanto la placa cromotrópica como la feromona deben reponerse periódicamente, cada unos cuantos meses (según indique el fabricante), ya que con el tiempo se desgastan y pierden efectividad.

El estudio de la población se realiza de forma semanal durante el periodo de observación. Se cuenta el número de moscas atrapadas cada día y se calcula el MTD (Ec. 2.1.) que es el número de moscas por trampa y día.

$$MTD = \frac{N^{\circ} de adultos capturados}{N^{\circ} de trampas * N^{\circ} de días transcurridos}$$

Ecuación 2.1. Mosca por Trampa y Día

Ejemplo: Hay 3 trampas colocadas y a lo largo de 1 semana de atrapan una serie de moscas (Tabla 2.1.).

Trampa Nº1	3 machos	1 hembra
Trampa Nº2	2 machos	0 hembras
Trampa Nº3	5 machos	1 hembra

Tabla 1. Ejemplo de capturas

De modo que el valor medio de capturas es:

$$MTD = \frac{12}{3 * 7} = 0.57 \frac{moscas}{día y trampa}$$

Cuando el MTD es superior a 0.6 es necesario tomar medidas contra la población de moscas [2].

Junto con el control de las trampas hay que realizar un muestreo de frutos con un diámetro de 10mm o más. El estudio consiste en recolectar 400 frutos de 20 árboles distintos en el caso del olivar de almazara y 1000 frutos de 20 árboles en el caso del olivar de mesa, realizando un conteo de cuantas aceitunas están picadas y cuáles no, además de otros índices.

% aceitunas con picada total: Porcentaje de frutos que tienen algún indicio de picada de *Bactrocera Oleae* (picada, huevo, larva... tanto vivos como muertos).

% aceitunas con formas vivas: Porcentaje de frutos que tienen o han tenido una o más formas vivas desarrolladas en ellos, esto incluye plagas distintas a la mosca [2].

2.2.5. Medios de lucha

El tratamiento a parcheo es el método más común para luchar con esta plaga; es un método de control de adultos, para evitar la picada en la aceituna, que consiste en pulverizar agua con un insecticida autorizado y proteína hidrolizada sobre el tercio superior del árbol orientado al sur. Este tratamiento se aplica al 25% de la plantación, por lo que su impacto medioambiental es moderado.

En el caso de que los frutos picados por la mosca superen el 7% se realiza una pulverización total (tratamiento larvicida) a la copa del olivo. Es un tratamiento más agresivo y solo debe usarse en los casos más extremos, ya que tiene un impacto mucho mayor [4].

Debido a esto es crucial poder detectar los aumentos de población a tiempo para poder realizar los tratamientos menos invasivos y permitir así conservar un ecosistema saludable.

2.3. Sistemas de detección automáticos

Por lo general la detección de plagas la llevan a cabo los propios agricultores o biólogos contratados por los mismos. El procedimiento consiste en colocar las trampas cromotrópicas y contar manualmente el número de moscas atrapadas.

Los últimos años, con la llegada de enfermedades bacterianas como la *Xylella fastidiosa* están proliferando los sistemas automáticos para la monitorización de cultivos y teledetección hiperespectral y térmica junto con sistemas de biocontrol, con el fin de conseguir detecciones tempranas y reducir el empleo de fitosanitarios químicos [5].

Siguiendo la línea de los sistemas automáticos para la monitorización de cultivos se encuentran empresas como ENVIRA, la cual desarrolla dispositivos para monitorizar cultivos (Fig. 2.12.). Estos miden la temperatura del aire y de la hoja, humedad, contenido de agua, conductividad del suelo... y envía los datos a una plataforma, permitiendo una monitorización constante del estado del cultivo.



Figura 2.12. NANOENVI AG

En cuanto a la detección de insectos, empresas como SEMIOS (Fig. 2.13.) disponen de cámaras trampa automatizadas para una gran variedad de cultivos (excluyendo olivo). Obtienen imágenes diarias y recuento de los insectos atrapados, información accesible desde el móvil que permite ahorrar tiempo y dinero al agricultor.



Figura 2.13. Cámara trampa SEMIOS

El prototipo de detección de plagas creado por José Luis Fernández Collado en la UJA como TFG (Fig. 2.14.), es capaz de detectar los insectos atrapados gracias a un sensor de presencia de movimiento FC-51 con un porcentaje de acierto del 60% y comunicar los resultados a una plataforma web. Es ejemplo de la constante investigación por mejorar la detección de plagas.



Figura 2.14. Trampa inteligente

El campo de detección de insectos se sigue desarrollando, ya que es de gran interés para la comunidad agrícola. Proyectos como FruitFlyNet-II, en el cual participa la UCO, cuentan con el apoyo económico de la unión europea y se centra en reconocer a la mosca del olivo a partir de imágenes digitales, utilizando técnicas de inteligencia artificial, con el fin de conseguir una temprana detección y reducir los químicos fitosanitarios empleados en el control de la plaga [8].

3. Hardware del prototipo

En este capítulo se desglosan todos los materiales necesarios para construir el sistema automático de detección de plagas en el olivar, justificando cada elección de cada componente según las características requeridas para el correcto funcionamiento del equipo.

3.1. Controlador

Funcionará como el cerebro del sistema, coordinando las acciones de los demás componentes para conseguir enviar las imágenes por correo. Dado que el sistema debe adquirir las imágenes es necesario utilizar un controlador familiarizado con las imágenes, de lo contrario tendrían que trabajarse 1 a 1 los píxeles de la hipermatriz resultante, por lo que se utilizará una placa Raspberry como controlador.

La placa Raspberry es un dispositivo electrónico muy utilizado en la industria. Este consta de un chip con arquitectura ARM, lo que lo hace más similar a un móvil que a un ordenador. Sin embargo, debido a su popularidad existen sistemas operativos completos para que funcione como un ordenador, aunque con una potencia mucho más limitada.

Como sistema operativo se utiliza Raspbian, Linux, el cual es el más extendido entre los usuarios de Raspberry y permite programar tareas e instalar una gran variedad de librerías y programas, lo cual es ideal a la hora de programar la captura de la imagen y su posterior envío por correo.

3.1.1. Requerimientos

Dado que se trata de un sistema autónomo, es imprescindible que el consumo sea lo más reducido posible durante su funcionamiento, de esta manera la autonomía del sistema será mayor. Este es el principal requisito, pero no el único, ya que es necesaria una potencia de cómputo moderada que permita realizar las tareas sin mucho esfuerzo.

A continuación, se muestran distintos parámetros técnicos de los distintos modelos Raspberry (Figuras 3.15. 3.16. y 3.17.):

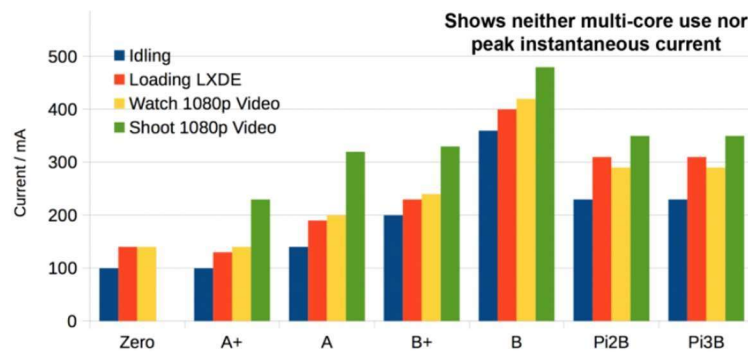


Figura 3.15. Consumo en funcionamiento

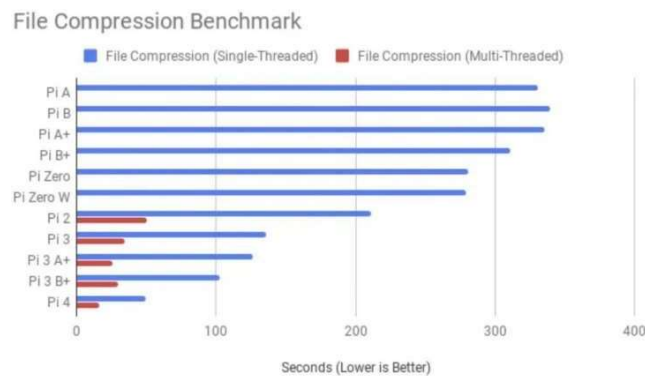


Figura 3.16. Compresión de archivos

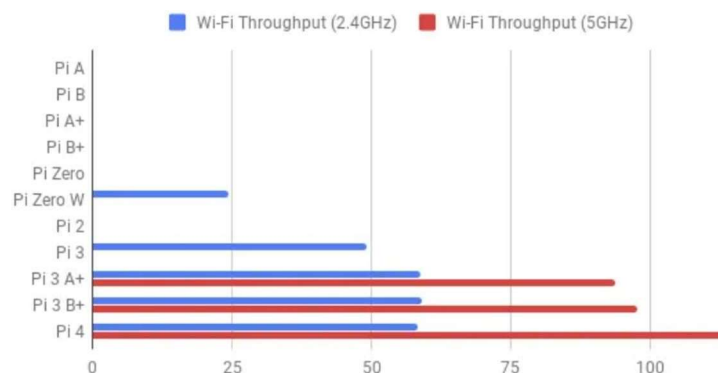


Figura 3.17. Velocidad de conexión Ethernet

Observando las anteriores gráficas se deduce que los modelos más modernos como son Pi3A+, Pi3B+ y Pi4 son los que ofrecen unas mejores capacidades de cómputo y tienen un consumo energético reducido, por lo que podrían cumplir de sobra con las condiciones anteriormente mencionadas.

Entre los 3 modelos mencionados, el modelo Pi3B+ se sitúa en un punto intermedio, en cuanto a potencia y consumo, pero destaca en regulación de temperaturas gracias a un disipador sobre el SoC (Fig.3.18.), esto podría ser decisivo para las olas de calor que suelen suceder los meses de verano.

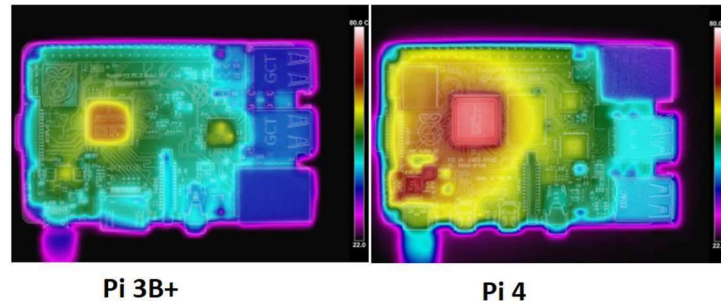


Figura 3.18. Temperatura en funcionamiento

Considerando todo lo anterior, el modelo Pi 3 B+ (Fig. 3.19.) es el más indicado para la tarea.



Figura 3.19. Raspberry Pi 3 B+

3.1.2. Optimización del consumo

Dejar en reposo la Raspberry resulta en un gasto de energía innecesario, por lo que apagarla una vez se ha completado la tarea permite reducir el consumo. Esta acción puede ejecutarse mediante el comando `-shutdown`, sin embargo, no hay forma de encender la Raspberry desde el código.

Linux es un sistema operativo que permite programar tareas, se puede realizar la adquisición y el envío de imágenes de forma automática cada vez que se encienda el dispositivo, pero no hay manera de que conecte la alimentación de la Raspberry estando apagado.

La Raspberry no consta de un RTC (Real Time Clock) que permita encender la Raspberry cuando sea necesario, por lo que es necesario un temporizador que realice esta acción.

Un PiWatcher (Fig. 3.20.) soluciona el problema. Este dispositivo está diseñado para reiniciar la Raspberry después de un cierto tiempo mediante el comando *PiWatcher wake (Time)*. Se conecta al hardware a través de los terminales señalados en la figura 3.17. “To Raspberry-Pi GPIO header” y permite reducir el consumo de los 200mA de la Raspberry en reposo 3mA.

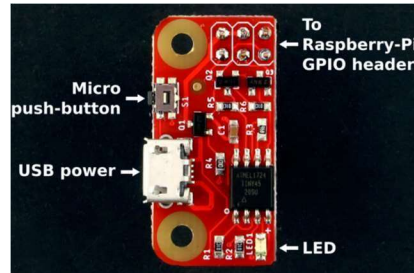


Figura 3.20. PiWatcher

Se trata de un dispositivo novedoso de difícil obtención. Dado que el sistema es muy simple no será necesaria su implementación, pero para proyectos futuros en los que se agreguen más dispositivos, convendría tenerlo en cuenta.

3.2. Conexiones red

Para llevar a cabo este proyecto es necesario que la Raspberry pueda acceder a internet y enviar así las imágenes por correo. Dado que el sistema se va a ubicar en mitad del campo, es necesario un punto de conexión a la red.

Se puede pensar en utilizar un router normal y corriente, pero presentan 2 problemas: el primero es la necesidad de una toma a 220v para su funcionamiento, lo cual daría lugar a un circuito de potencia más complejo. El segundo problema es la conexión del dispositivo a internet. Para que un router común pueda acceder a internet es necesario configurarlo a través de una compañía móvil, lo cual supone un coste adicional durante la etapa de pruebas del dispositivo.

Teniendo en consideración lo anterior resulta conveniente analizar las alternativas, como son los router portátiles (Fig. 3.21.). Este dispositivo ofrece una conexión wifi a partir de una Red 4G, por lo que es posible obtener conexión a internet a partir de una tarjeta SIM con datos. Inicialmente se utilizará una tarjeta de prepago, pero para su uso prolongado es recomendable contratar una tarifa de datos de al menos 72Mb al mes, que es el peso de todas las imágenes que se mandarían.



Router

Figura 3.21. Conexiones red

Entre los distintos modelos de router portátiles hay que elegir aquel que presente un menor consumo, por lo que modelos con pantallas a color o táctiles tendrán un consumo extra innecesario. El modelo elegido es Tenda Router 4G móvil 4G180.

Para configurar la conexión a internet se debe configurar el router tras introducir la tarjeta SIM por primera vez. Para ello se conecta un dispositivo al wifi Tenda-4G y se busca por internet: 192.168.0.1 lo cual redirecciona a una página en la que debe escribirse el nombre y contraseña de la red wifi, en este caso el nombre elegido es Tenda-4G y la contraseña 123456789.

3.3. Sistema de Visión

Es el componente más importante de todo nuestro sistema, ya que se encarga de capturar la información que más adelante será procesada por el software, por lo que se deben cumplir una serie de requerimientos con el fin de garantizar el correcto funcionamiento de la misma una vez instalada.

3.3.1. Requerimientos climáticos

Dado que la cámara se va a situar en el campo, debe poder soportar las condiciones meteorológicas de la zona, estos dispositivos son sensibles y deben estar bien protegidos para evitar posibles daños y garantizar su durabilidad.

A continuación, se muestra un promedio del clima en los últimos 40 años en la provincia de Jaén (Figuras 3.22. y 3.23.).

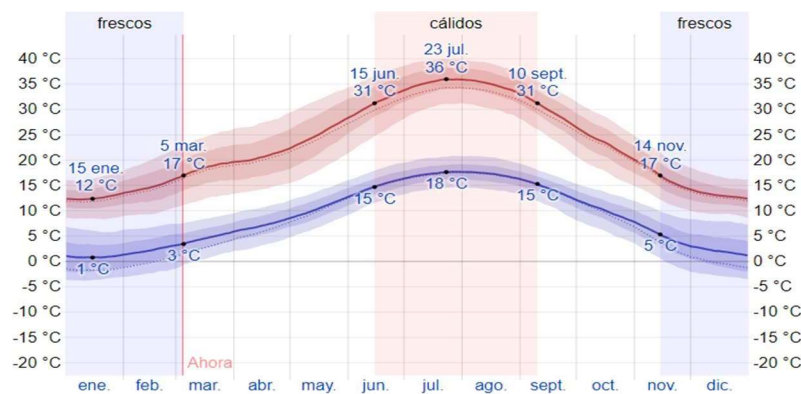


Figura 3.22. Temperaturas anuales en Jaén



Figura 3.23. Precipitaciones anuales en Jaén

En la provincia de Jaén, las temperaturas oscilan entre los 1°C y los 36°C, las precipitaciones son escasas, de modo que una protección simple (Figura 3.24.) que cubra la cámara para evitar la lluvia y la radiación directa es suficiente.

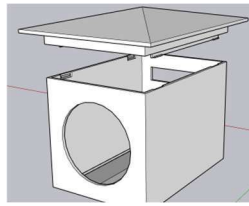


Figura 3.24. Diseño caja protectora 3D

Uno de los factores fundamentales en los sistemas de visión es la iluminación. Para conseguir una buena imagen es necesario que la zona de estudio esté bien iluminada, y dado que el sistema se va a situar al aire libre, es necesario saber si hará falta una iluminación especial a parte de la ambiental (Fig. 3.25. y Fig. 3.26.).

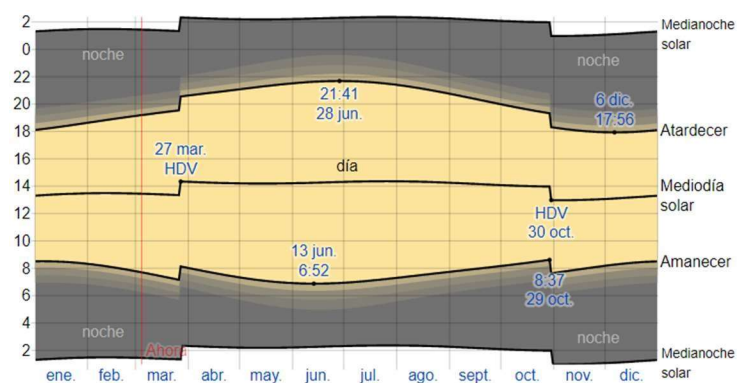


Figura 3.25. Salida y puesta del sol en Jaén

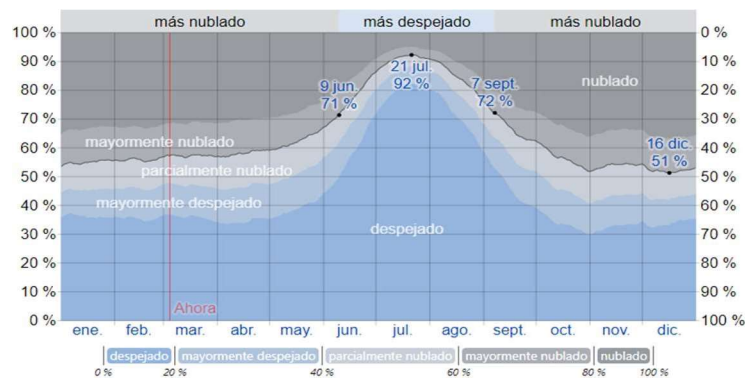


Figura 3.26. Nubosidad anual en Jaén

La provincia de Jaén tiene un clima bastante soleado y con muchas horas de luz al día, por lo que no sería necesario ningún tipo de iluminación extra, en todo caso, podrían realizarse varias fotografías al día para contrastar unas imágenes con otras, realizar un promedio de las moscas detectadas y garantizar así una lectura lo más correcta posible.

3.3.2. Requerimientos de visión por computador

Resulta especialmente importante analizar estos requerimientos, ya que permitirán obtener una imagen con la suficiente resolución para su posterior análisis. Entre los mismos cabe destacar el campo de visión a adquirir y los elementos a detectar.

Campo de visión: Es la región de interés, la cual enfoca la cámara y se obtienen imágenes. En este caso es una trampa cromotrópica (Fig. 3.27.), la cual tiene un tamaño similar al de un folio A4, sus dimensiones son de 250mm x 200mm. Es de color amarillo y está recubierta de una sustancia pegajosa.

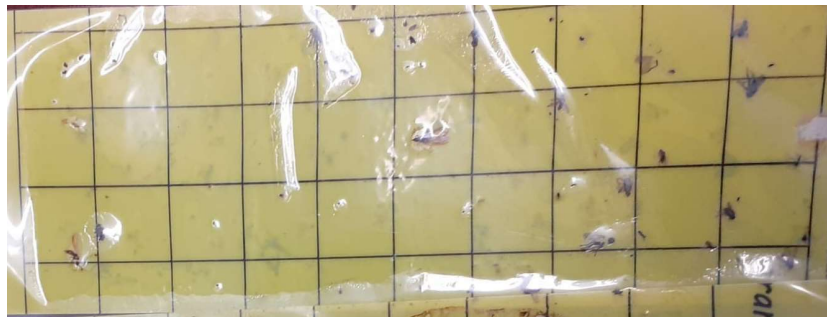


Figura 3.27. Trampa cromotrópica usada

Dada la forma tan alargada que tiene el campo de visión existen 2 opciones para captar la imagen:

-Colocar una única cámara (Fig. 3.28.) de manera que el borde largo de la trampa esté orientado con el lado con más píxeles de resolución de la cámara. Esta configuración permite optimizar el número de cámaras necesarias, reduciendo el presupuesto, pero se desaprovecha parte de la imagen captada.

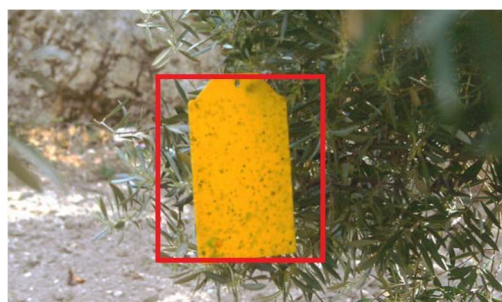


Figura 3.28. Campo de visión con 1 cámara

-Colocar 2 cámaras seguidas (Fig. 3.29.) ajustando el borde corto con el lado con mas píxeles de la cámara. De esta manera se optimiza mucho mejor la captura de la imagen y no se desaprovecha parte de la misma, consiguiendo aumentar al mismo tiempo la resolución de la imagen, sin embargo, se duplica el presupuesto en cámaras y requiere de una posterior alineación entre las imágenes capturadas.

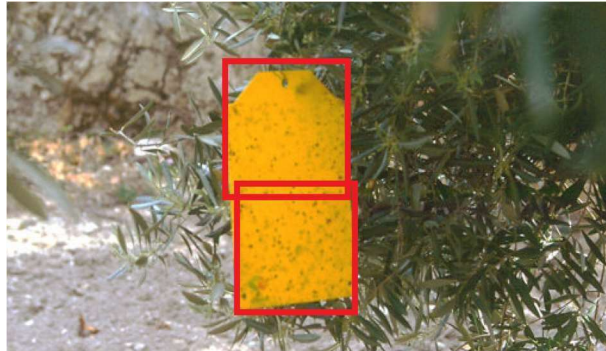


Figura 3.29. Campo de visión con 2 cámara

Elementos a detectar: El objetivo es detectar la mosca del olivo (Fig. 3.30.). Sus características más destacables son las siguientes:

- Longitud de 4mm a 5mm.
- Mancha negra en la punta de sus alas.
- Cuerpo con varios tonos marrones y un triángulo amarillento sobre el dorso y otros dos en la base de las alas.
- La hembra es algo más grande y tiene un ovipositor muy visible.



Figura 3.30. Mosca del olivo

Teniendo en cuenta estas características del insecto, es requisito tener una resolución aproximada a 10 píxeles/mm con el fin de poder identificar claramente las manchas amarillentas en el dorso o una resolución mayor para detectar la mancha negra de la punta de sus alas.

3.3.3. Elección de cámara

La función de la estación es tomar unas cuantas imágenes en las horas centrales del día, por lo que la cadencia de disparo de la cámara no cobra importancia a la hora de elegir la misma. Lo que prima la resolución máxima que puede adquirir la cámara con el fin de poder identificar los detalles que identifican a cada insecto.

Con el fin de analizar la resolución que puede alcanzarse con una cámara se aplica el modelo pin-hole (Fig. 3.31.), el cual permite saber la relación entre la superficie estudiada y la imagen obtenida mediante triángulos semejantes (Ec. 3.2.).

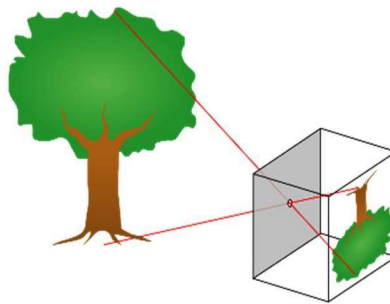


Figura 3.31. Modelo Pin-Hole

$$\frac{Area_{Estudio}}{Distancia} = \frac{Area_{Sensor}}{focal}$$

Ecuación 3.2. triángulos semejantes

Dado que se va a utilizar una Raspberry Pi 3 B+ hay que considerar los tipos de conexiones disponibles para la cámara (Fig. 3.32.): un puerto 2-lane MIPI CSI camera port y 4 puertos USB.

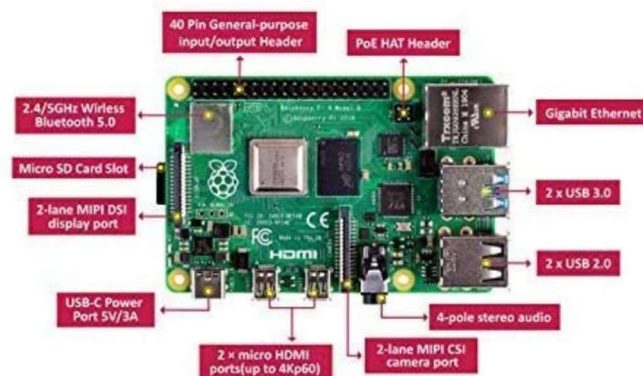


Figura 3.32. Conexiones Raspberry

Con el fin de que en futuros proyectos puedan incorporarse más cámaras, el método de conexión de esta con la Raspberry será el USB, ya que la Raspberry cuenta con 4 puertos y resulta muy sencillo implementar las cámaras de esta manera.

Cámara de vigilancia ELP 8MP (Fig. 3.33)

Esta cámara es capaz de funcionar entre -20°C y 70°C con una carcasa robusta, por lo que es capaz de resistir las condiciones meteorológicas de la zona.



Figura 3.33. Cámara ELP 8MP

Tiene una resolución máxima de 3264 x 2448 pixeles y un tamaño de pixel de $1,93 \cdot 10^{-3}mm$ y una focal variable de entre 2 y 12 mm. Con estos valores se calcula la resolución para cada uno de los modos de trabajo

Para la configuración de una sola cámara de la Figura 3.33., aplicando la ecuación 3.1, la resolución obtenida es la siguiente:

$$Distancia = \frac{focal \cdot Area_{Estudio}}{Area_{Sensor}} = \frac{297 \cdot 12}{1,93 \cdot 10^{-3} \cdot 3264} = 476.2mm$$

$$Resolución = \frac{Area_{Estudio} \cdot focal}{Distancia \cdot tamaño_{pixel}} = \frac{1 \cdot 12}{565.7 \cdot 1,93 \cdot 10^{-3}} = 13.06 \text{ pixl/mm}$$

En el caso de la configuración de doble cámara, Figura 3.33., la resolución obtenida es de:

$$Distancia = \frac{focal \cdot Area_{Estudio}}{Area_{Sensor}} = \frac{200 \cdot 12}{1,93 \cdot 10^{-3} \cdot 3264} = 381mm$$

$$Resolución = \frac{Area_{Estudio} \cdot focal}{Distancia \cdot tamaño_{pixel}} = \frac{1 \cdot 12}{381 \cdot 1,93 \cdot 10^{-3}} = 16.3 \text{ pixl/mm}$$

Dado que se superan los 10 pixeles/milímetro, esta cámara cumple todos los requisitos necesarios y puede utilizarse en cualquiera de los 2 modos. Dado que la diferencia de resolución es mínima se utilizará la configuración de 1 cámara para ahorrar en el presupuesto.

3.4. Fuente de alimentación

Debido a la falta de infraestructuras en la zona de trabajo, la forma más sencilla y ecológica de alimentar el sistema es mediante una instalación fotovoltaica, aprovechando así los altos niveles de radiación solar que tiene la provincia de Jaén.

Una instalación fotovoltaica consta de una placa solar, que convierte la radiación en energía eléctrica, un regulador de carga como sistema de protección y una batería para asegurar la autonomía del sistema. El dimensionamiento del equipamiento se expone a continuación.

3.4.1. Consumo del sistema

En primera instancia es necesario analizar el consumo de todos los componentes del sistema con el fin de tener una idea de la energía necesaria para el funcionamiento del mismo:

-Raspberry: Trabaja durante 24 horas al día, en las cuales toma las fotografías y las envía por correo. Tras realizar un estudio mediante el uso de un enchufe inteligente se pudo medir el consumo diario, el cual es de 50Wh/día.

-Router wifi: Está constantemente encendido, 24h/día, consume 3.8V y 210mAh.

Teniendo en cuenta el consumo de cada componente por separado se puede calcular el consumo diario total (Ec. 3.3.):

$$E = \sum V \cdot I \cdot t$$

Ecuación 3.3. Cálculo del consumo diario

De modo que el consumo diario del sistema de detección de plagas es de:

$$E = 50 + 3.8 \cdot 0.21 \cdot 24 = 69 \text{ wh/día}$$

Con el fin de garantizar el suministro de energía, los cálculos de aquí en adelante se realizarán sobredimensionando el sistema y considerando un consumo diario de 70wh.

3.4.2. Dimensionamiento de la placa fotovoltaica

Conocido el consumo del sistema de detección (70wh/día) es necesario averiguar el índice de radiación solar de Jaén, el cual podemos consultar en la Agencia Andaluza de la Energía (Fig. 3.34.).

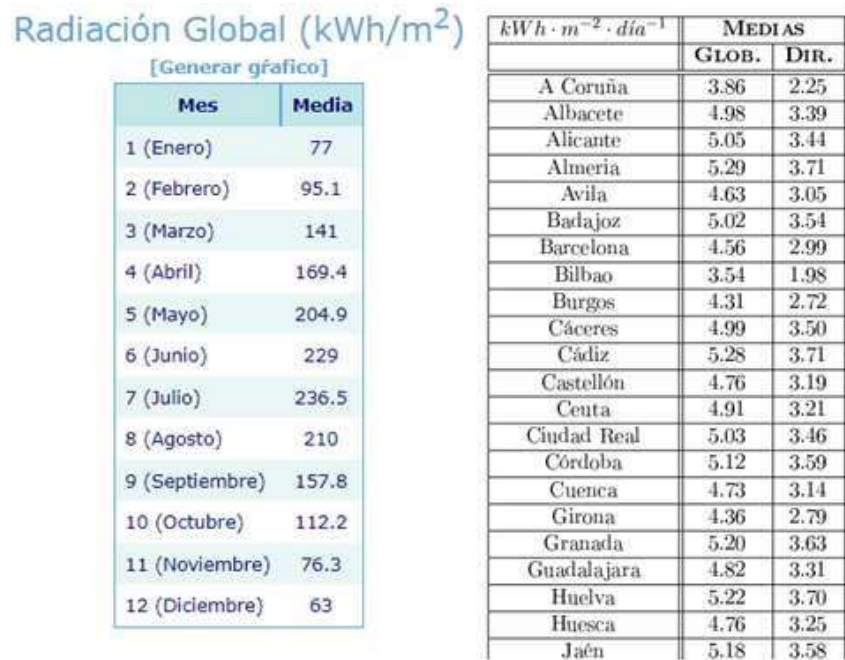


Figura 3.34. Radiación por la Agencia Andaluza y por el atlas de la radiación.

Calculando el promedio de los valores mensuales de radiación global horizontal ofrecidos por la Agencia Andaluza de la Energía se obtiene un $H=6.1$, mientras que el atlas de la radiación muestra una $H=5.18$, por lo que se utilizará un valor intermedio.

$$H = 5.8 \text{ Kwh/m}^2/\text{día}$$

Sabiendo el valor de H solo queda aplicar la ecuación 2 para hallar la potencia necesaria a instalar (Ec.3.4.):

$$E = PR \cdot P \cdot \frac{H}{G}$$

Ecuación 3.4. Potencia que instalar

De modo que la potencia necesaria es:

$$70 \text{ wh/día} = 0.8 \cdot P \cdot \frac{5.8}{1}$$

$$P = 15.09 \text{ W}$$

Se debe instalar una potencia superior o igual a 15W, por lo que un módulo fotovoltaico de 15W (Fig. 3.35.) es suficiente para alimentar el sistema de detección de plagas.



Figura 3.35. Módulo fotovoltaico de 15W.

3.4.3. Dimensionamiento de la batería

El sistema debe contar con energía por la noche, momento en el cual las placas no aportan energía, además hay que tener en cuenta que pueden pasar varios días seguidos con el cielo nublado y no generar suficiente energía, por lo que la batería es un componente imprescindible para asegurar el correcto funcionamiento del sistema.

Teniendo esto en cuenta, se dimensiona la batería para que pueda aguantar 2 días seguidos sin cargarse en absoluto y procurando que la carga no descienda de 50%, esto último con el fin de optimizar los ciclos de carga y descarga, aumentando así la vida útil de la batería (Ec. 3.5.).

$$Capacidadbatería = \frac{N^{\circ}díasdeautonomía \cdot Consumodiario}{Máximaprofundidaddescarga}$$

Ecuación 3.5. Capacidad de la batería

Aplicando la ecuación anterior se obtiene la potencia necesaria que debe almacenar la batería.

$$Capacidadbatería = \frac{2 \cdot 70}{0.5} = 280W$$

Considerando el valor estándar de 12V, la batería debe tener una capacidad de 23,3Ah para satisfacer las condiciones, considerando que la radiación solar realmente nunca llega a 0 y que el sistema se utilizará en los meses de mayo-noviembre, una batería de 18Ah (Fig. 3.36.) es suficiente para alimentar el sistema.



Figura 3.36. Batería 12V y 18Ah

3.4.4. Regulador de carga

Se trata de un elemento cuya función es controlar el estado de la carga de las baterías, ofrecen protección contra cortocircuitos, circuito abierto y corriente inversa entre otros, lo cual resulta imprescindible para los sistemas fotovoltaicos.

En este caso es necesario que pueda trabajar a 12 Voltios y 1.2 Amperios, como mínimo. También hay que considerar las salidas USB del regulador, ya que en estas se conectarán la Raspberry y el router. Dicho USB debe ser capaz de entregar 5V y 3A. Teniendo todo esto en cuenta, un modelo de 12V y 30A (Fig. 3.37.) cumple los requisitos.



Figura 3.37. Regulador de carga

3.5. Montaje del sistema

Una vez se tienen todos los componentes es necesario conectarlos y dotarlos de una estructura que soporte y proteja el sistema completo. Para ello se diseña una estructura mediante SkechUp (Fig. 3.38.). Dicha estructura debe elevar los componentes con respecto al suelo y cubrirlos para evitar la adversa meteorología a la vez que sitúa la cámara y la trampa a una distancia constante.

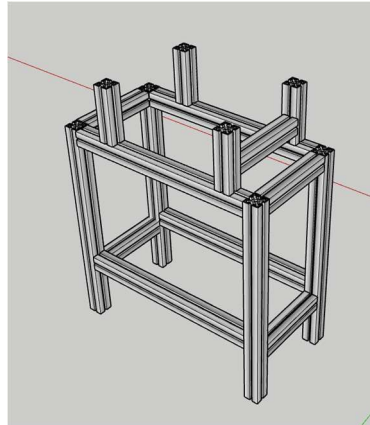


Figura 3.38. Estructura SkechUp

Teniendo clara la idea de la estructura se buscó con qué materiales hacerse. En este caso se optó por perfiles FASTEN 30x30, ya que el grupo GRAV disponía de una gran cantidad de estos perfiles en su despacho y se decidió reciclarlos para este proyecto.

Dado que los perfiles tenían longitudes y cortes indeseados, se procedió a mecanizar el material para darle la forma oportuna mediante el uso de la maquinaria del laboratorio de mecánica. Se cortaron las medidas correctas y se taladraron los perfiles para poder atornillarse unos a otros.

Una vez construido el esqueleto se necesitaban unas paredes, para lo cual se utilizaron chapas de aluminio en la parte superior e inferior y madera para los laterales, materiales que iban a ser desechados y, tras recortarlos con las medidas deseadas, fue posible darles una segunda vida.

Gracias a todos los materiales reciclados y reacondicionados para la construcción de la estructura (Fig. 3.39.) se optimizaron los recursos empleados, fomentando el desarrollo sostenible.



Figura 3.39. Estructura

La parte interior contiene la Raspberry, el router, la batería y el regulador de carga. El diseño cerrado evita que entre agua e incida directamente la radiación solar. Es posible acceder a la parte interior desatornillando el cierre metálico lateral.

De la parte interior salen 2 cables, uno se dirige a la placa solar, la cual se encuentra en el lateral de la estructura y será orientada al sur con una inclinación cercana a los 20º con el fin de optimizar la adquisición de energía durante su funcionamiento. El otro cable corresponde con la cámara, que se sitúa sobre el apoyo trasero cubierta con la protección 3D.

La trampa cromotrópica se sitúa justo delante, a una distancia de 40 cm, fijada con una cuerda para evitar que se mueva y permitiendo que el operario pueda cambiar sencillamente las trampas cuando requieran feromonas o dejen de pegar.

4. Software del prototipo

En este capítulo se explican todos los pasos a seguir para conseguir el correcto funcionamiento del sistema desde el punto de vista de la programación. Se muestra tanto la programación de la Raspberry como la programación del software de reconocimiento.

4.1. Programación Raspberry

El objetivo de la Raspberry es dar la orden de adquirir las imágenes de la trampa cromotrópica y enviarlas por correo para su posterior análisis. Inicialmente la Raspberry no tiene ningún software instalado, por lo que el primer paso es instalarle un sistema operativo.

4.1.1. Instalación y puesta a punto del sistema operativo

El sistema operativo por instalar es Raspbian, un sistema operativo GNU/Linux basado en Debian, el cual puede descargarse desde la página oficial de Raspberry. Este sistema operativo es compatible con todos los modelos de Raspberry y es el más utilizado por la comunidad.

Existen 3 versiones de Raspbian, una con varios softwares instalados, otra con lo básico y una versión reducida que no tiene interfaz gráfica (se controla desde el terminal). Para este proyecto, se utiliza la versión con lo básico (Fig. 4.40.) y se instalarán los paquetes que sean necesarios.



Figura 4.40. Versión básica de Raspbian

Una vez descargado Raspbian es necesario instalarlo en una tarjeta SD como sistema operativo. Existen muchas formas de realizar la partición de la tarjeta SD para conseguir este objetivo, como por ejemplo utilizar BalenaEtcher (Fig. 4.41.), software especializado en esto y muy sencillo de utilizar.

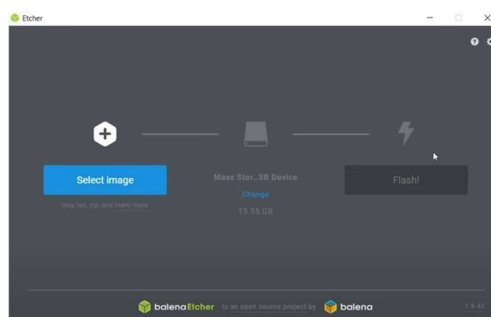


Figura 4.41. BalenaEtcher

Tras instalar el sistema operativo, solo queda introducir la tarjeta a la placa Raspberry y conectar los periféricos (Ratón, teclado, pantalla...). La primera vez que inicia Raspbian se debe configurar el idioma, la hora, la resolución de pantalla y la conexión de red. Tras configurar se reinicia el sistema automáticamente.

Una vez reiniciado, el primer paso es instalar las actualizaciones del sistema, para lo cual debe haber una conexión red activa, y se realiza ejecutando los siguientes comandos desde el terminal:

<code>sudo apt-get update</code>	Busca las actualizaciones del sistema.
<code>sudo apt-get upgrade</code>	Instala las actualizaciones encontradas.

Tras unos minutos el sistema está listo para utilizarse [9].

4.1.2. Conexión y configuración de la cámara

La cámara se conecta a uno de los terminales USB de la Raspberry y ya estaría disponible para su uso. Para que el software de Raspbian reconozca la cámara es necesario instalar una herramienta que nos permita establecer conexión con la misma y acceder a sus funciones. La herramienta que se va a utilizar es fswebcam y se instala desde el terminal con el siguiente comando:

```
sudo apt install fswebcam
```

La herramienta fswebcam incluye una gran cantidad de comandos con los que controlar los parámetros de adquisición de imágenes, entre los que cabe destacar:

fswebcam imagen.jpg	captura una imagen y la guarda con el nombre imagen.
fswebcam -r 1280x720	cambia la resolución de la cámara a 1280x720.
fswebcam -s 3	salta los 3 primeros fotogramas adquiridos.
Fswebcam -F 2	Combina 2 fotogramas consecutivos
fswebcam -help	Permite ver una lista con todos los parámetros.

Combinando estas instrucciones pueden realizarse capturas de imagen correctamente (Fig. 4.42.). Es necesario saltar los primeros fotogramas, pues son los que se utilizan para ajustar el nivel de brillo.

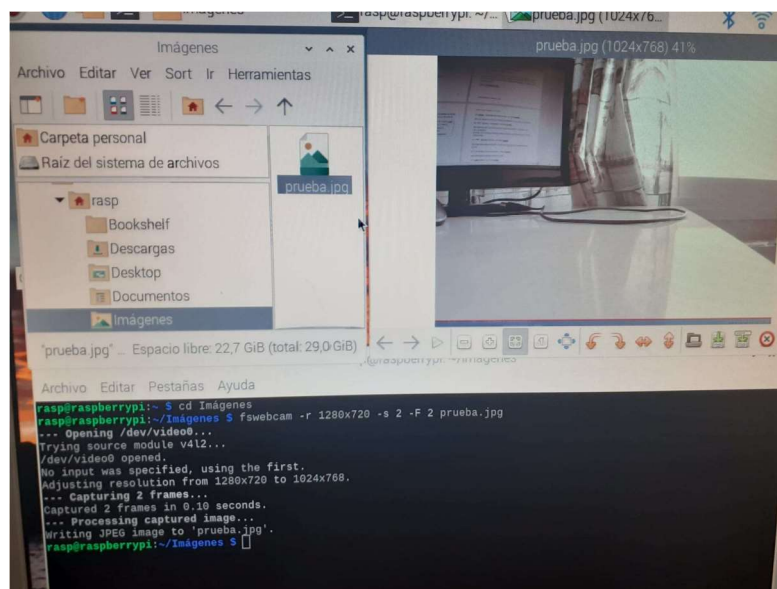


Figura 4.42. Prueba de captura de imagen

Para conseguir la máxima resolución de la cámara se puede consultar un listado de las resoluciones disponibles con el comando:

```
V4l2-ctl -list-formats-ext | more
```

En nuestro caso concreto la máxima resolución disponible se adquiere con el comando:

```
fswebcam -r 3640x2448 -pYUYV
```

4.1.3. Rutina de adquisición y envío de imágenes

Una vez conocemos como trabajar con la cámara es necesario crear una rutina que realice la captura de la imagen y la envíe por correo periódicamente. Para programar esta rutina se utilizará Python, el cual es el lenguaje más común para programar en Raspberry.

El primer punto es instalar Python en el dispositivo, lo cual puede hacerse desde el Terminal ejecutando el siguiente comando:

```
Sudo apt install python3 idle3
```

Python cuenta con un gran repertorio de librerías, entre las cuales nos quedaremos con las siguientes para realizar nuestra rutina:

os	Permite lanzar un comando para ejecutarse en el terminal
smtplib	Módulo que permite el envío de correos
time	Permite extraer la fecha

El código rutina.py está estructurado de la siguiente manera (Diagrama 4.1.):

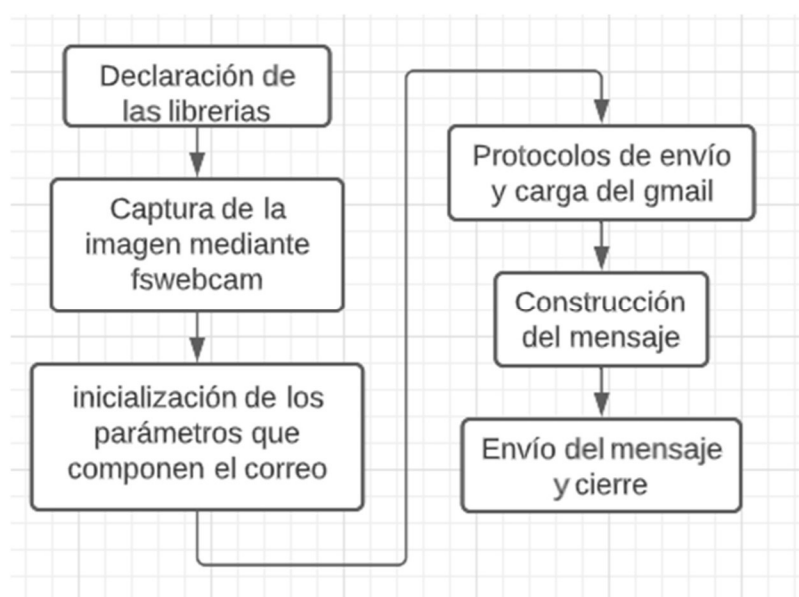


Diagrama 4.1. Rutina.py

#rutina.py

```
import time
import os, smtplib, getpass
from datetime import datetime

fecha=datetime.today().strftime('%d/%m/%Y')
os.system('fswebcam -r 3264x2448 -pYUYV --skip --no-banner -F 2 --save
/home/rasp/Imágenes/Trampa.jpg')
time.sleep(2)

from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
from email.mime.base import MIMEBase
from email.encoders import encode_base64
user='correo@gmail.com'
password='contraseña'
remitente='remitente@gmail.com'
destinatario='daniel <detectamoscas@gmail.com>'
asunto='Imagen de la trampa'
mensaje=fecha
archivo='/home/rasp/Imágenes/Trampa.jpg'

gmail=smtplib.SMTP('smtp.gmail.com',587)
gmail.starttls()
gmail.login(user,password)

gmail.set_debuglevel(1)
cabecera = MIMEMultipart()
cabecera['Subject']=asunto
cabecera['From']=remitente
cabecera['To']=destinatario
mensaje=MIMEText(mensaje, 'plain')
cabecera.attach(mensaje)
if (os.path.isfile(archivo)):
    adjunto=MIMEBase('application','octet-stream')
    adjunto.set_payload(open(archivo,"rb").read())
    encode_base64(adjunto)
    adjunto.add_header('Content-Disposition','attachment; filename="%s" '
% os.path.basename(archivo))
    cabecera.attach(adjunto)

#envio y cierre
gmail.sendmail(remitente, destinatario, cabecera.as_string())
gmail.quit()
```

Cabe destacar que el correo desde el cual se realiza el envío debe estar configurado para trabajar con el protocolo smtp. En el icono configuración de Gmail debe estar habilitada la opción: Reenvío y correo POP/IMAP -> Habilitar IMAP.

Además, se debe configurar la Cuenta. En el apartado de Seguridad debe estar activa la verificación en 2 pasos y posteriormente crear una contraseña de aplicación, la cual se utilizará para acceder al correo desde la raspberry.

Tras ejecutar el código de la rutina aparece el siguiente mensaje en el Gmail del destinatario (Fig. 4.43.), el cual contiene la imagen de la trampa y la fecha en la que se ha realizado la captura [10].

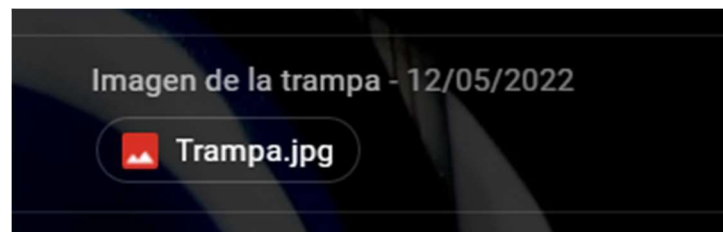


Figura 4.43. Mensaje en la bandeja de entrada Gmail

Ahora solo queda programar que esta rutina se ejecute de forma periódica. Cron y Crontab son herramientas incluidas en los sistemas Linux como Raspbian que permiten ejecutar tareas en un momento concreto. Cron está constantemente ejecutándose desde el momento que se enciende la Raspberry, comprobando si tiene alguna tarea para ejecutar, mientras que Crontab es un archivo de texto en el que se recogen las tareas que deben ser ejecutadas y el momento en el que deben ejecutarse.

Para agregar tareas a Crontab se debe ejecutar el comando “crontab -e” en el terminal, si es la primera vez que se ejecuta preguntará que editor de texto desea usarse. La opción /bin/nano es la más sencilla de trabajar (Fig. 4.44.).

```

# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command

```

Figura 4.44. Crontab

Todas las líneas que empiezan con # corresponden a comentarios, por lo que para programar una tarea se debe bajar hasta la última línea del archivo. Es un proceso muy sencillo, primero se escribe el momento en el que se desea la ejecución de la tarea y posteriormente el comando que se desea ejecutar. Tras escribirlo, se guarda el archivo crontab con [Ctrl+O, Enter] y se sale del editor con [Ctrl+X].

La configuración del momento se compone de 5 instrucciones (minuto, hora...) separadas con espacios (Fig. 4.45.). Dentro de cada instrucción pueden colocarse distintos símbolos (*, /, - ...) para obtener el momento exacto en el que se desea la ejecución.

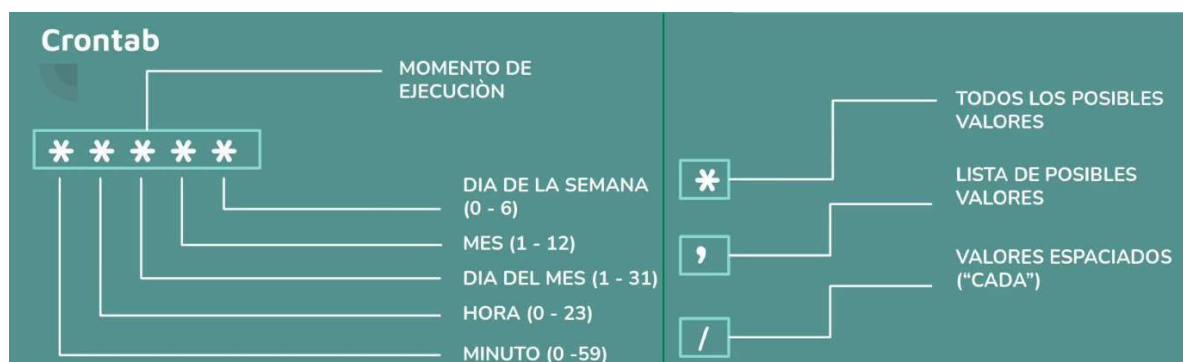


Figura 4.45. Configuración del momento de ejecución Crontab

Dado que la creación del momento de ejecución puede dar lugar a confusión, se recomienda consultar el momento creado en la web “crontab guru” (Fig. 4.46.), la cual escribe en texto aquello que pongamos como instrucción. En el caso del detector de plagas, es interesante realizar varias capturas al día en las horas de mayor iluminación por si alguna de ellas saliese mal. Esta acción debería realizarse todos los días mientras se esté realizando el estudio [11].

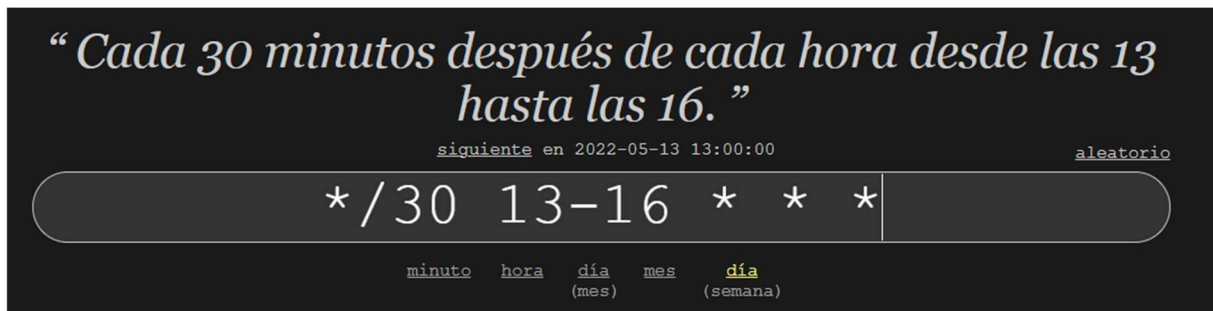


Figura 4.46. Crontab guru

De modo que la instrucción final quedaría como:

```
* /30 13-16 * * * python3 rutina.py
```

4.2. Detector de plagas

Una vez el usuario recibe la imagen de la trampa por correo ya podría realizar un conteo manual de la cantidad de moscas que han sido atrapadas, sin embargo, a continuación, se propone un software que permiten realizar un conteo automático del número de moscas mediante técnicas de visión por computador.

Existen varios entornos de programación en los que realizar el software, Python, visual studio, pycharm... En este caso se utilizará Matlab debido a la gran variedad de aplicaciones que tiene instalado (ImageAdquisition, Camera Configuration...).

4.2.1. Obtención de imágenes

El primer paso para poder crear un algoritmo de visión es la obtención de imágenes con las que se trabajará. Es importante tener una amplia base de datos para poder aplicar las oportunas técnicas e identificar correctamente a las moscas.

En este caso se trabajó insectos capturados en 2 recipientes en los cuales estaban conservados en una solución de 80% etanol y 20% agua destilada (Fig. 4.47.). Esta solución hace que pierdan color con el tiempo, pero dado que eran capturas recientes, las moscas aún conservaban parte de las tonalidades marrones y manchas blancas que las caracterizan.



Figura 4.47. Insectos capturados

Además de las moscas del olivo también hay diferentes especímenes, como hormigas, moscardones, pulgones... con el fin de poder entrenar distintos algoritmos para hallar las diferencias que existen entre los insectos que queremos detectar y los que no.

La idea es ir colocando los insectos sobre un folio amarillo que simula ser la trampa cromotrópica, de esta manera se pueden obtener un número ilimitado de imágenes sin necesidad de esperar a que se capturen en varias trampas.

Se debe tener en cuenta la naturaleza de la trampa, estará colocada en mitad del campo, por lo que los insectos pueden caer en una gran variedad de posiciones, sin embargo, el caso de los insectos voladores es un poco diferente. Estos intentarán apoyarse sobre sus patas para aterrizar en la superficie de la trampa, aunque no siempre lo consigan, por lo que lo más probable es que se queden pegados sobre sus patas o de costado.

A la hora de proceder, el primer paso es sacar los distintos especímenes de la disolución con ayuda de una pinza quirúrgica y colocarlos sobre un papel para que se sequen y recobren su solidez. A continuación, se calibra la distancia focal y el enfoque de la cámara para obtener la mejor resolución posible. Finalmente se puede comenzar con la captura de imágenes (Fig. 4.48.).

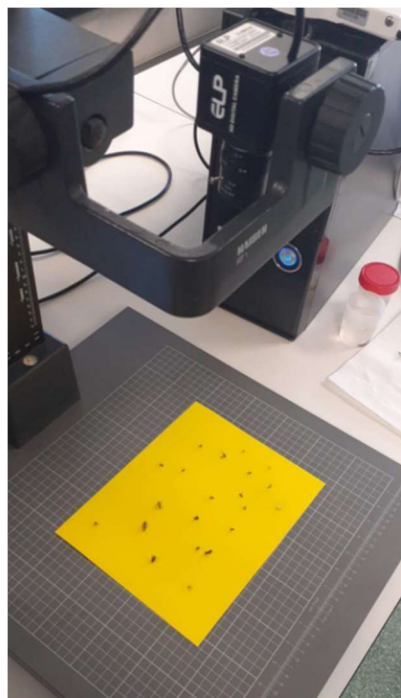


Figura 4.48. Adquisición de imágenes

El objetivo ahora es identificar la mosca del olivo de entre los insectos que estén atrapados en la Trampa (Fig. 4.49.). Para ello se utilizará un algoritmo de clasificación basado en clasificación mediante redes neuronales, el cual se explica a continuación.



Figura 4.49. Distintos insectos capturados

4.2.2. Clasificación mediante redes neuronales

Este método consiste en identificar las distintas regiones que hay dentro de la imagen, extraer un vector de características las cuales sean lo más discriminadoras y fiables posibles, y entrenar una red neuronal con dicho vector para clasificar cada insecto como mosca o no mosca.

El primer punto para conseguir dichas características es el preprocesado de la imagen. La cámara tiene una distorsión de barril que manipula las dimensiones reales de la trampa (Fig. 4.50.), por lo que es necesario revertir este efecto.



Figura 4.50. Distorsión de barril

Matlab cuenta con una herramienta llamada “Cámara Calibrator”, la cual genera una matriz de deformaciones que permite revertir el efecto de la lente. Para utilizar esta herramienta es necesario fotografiar un tablero de cuadros con la cámara en cuestión desde varios puntos e introducir la medida de los cuadrados. La aplicación mide la distancia entre las esquinas de los cuadrados y reconstruye la escena (Fig.4.51.) para determinar cuál ha sido la distorsión provocada por la cámara.

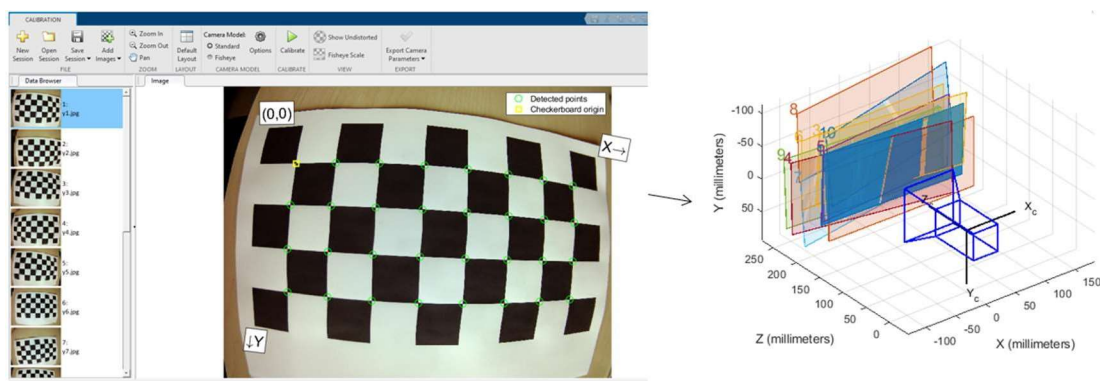


Figura 4.51. Camera Calibrator

Con el fin de extraer las regiones que delimitan a los distintos insectos se utiliza un algoritmo de umbralización. Estos algoritmos permiten convertir una imagen en escala de grises en una imagen binaria, donde los 1 representan los puntos de interés y los 0 el fondo. Dada la variedad de colores que presentan los insectos y la alteración de la iluminación provocada por el entorno natural de la trampa.

Se opta por realizar una umbralización manual a través de un Slider. Se regulará el Umbral procurando obtener la mayor distinción de los insectos sin llegar a saturar la imagen (Fig. 4.52.). Es un método poco robusto y escalable, sin embargo, permite identificar los insectos con un tamaño similar al real. Tras la aplicación del umbral, es necesario invertir la matriz resultante para tener como 1 (blanco) las regiones de interés.

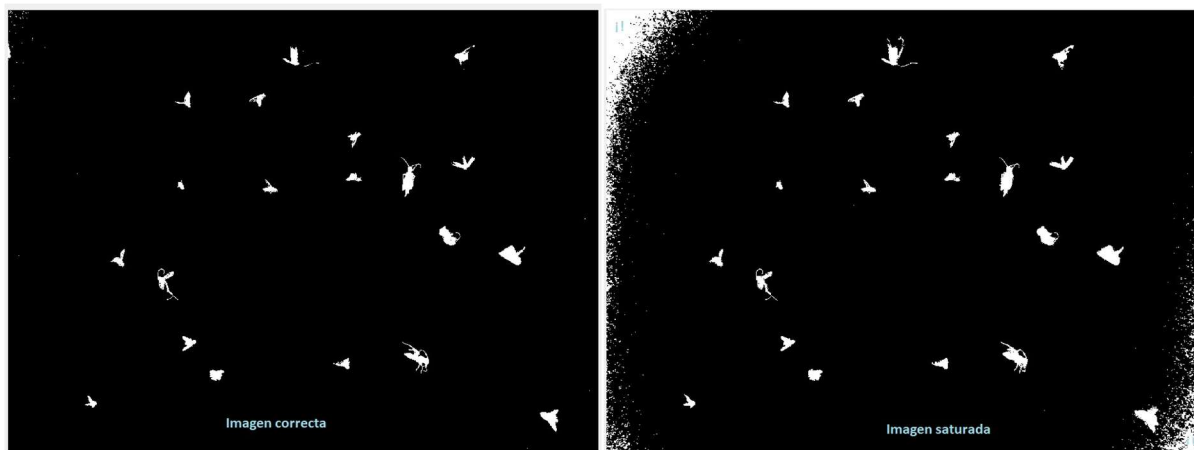


Figura 4.52. Imagen segmentada manualmente

Ahora que los insectos están segmentados es posible identificar cada insecto de la imagen, sin embargo, esta segmentación no sigue una metodología y no permite extraer características correctamente debido a la elección arbitraria del usuario del umbral. Se obtiene una buena detección del insecto, pero una mala delimitación del mismo.

Para arreglar este problema, se aprovecha la buena detección del insecto para extraer un recorte, sobredimensionarlo para incluir partes del insecto que hayan podido quedar fuera y umbralizar esta nueva región con un método distinto. Concretamente se utiliza el algoritmo graythresh, el cual calcula el valor del umbral mediante el método Otsu, que consiste en buscar un valor umbral el cual dé lugar a la menor varianza posible del nivel de gris dentro de las regiones delimitadas.

Tras esto se pueden extraer las características geométricas más identificativas de los insectos. Se debe tener en cuenta que las características deben separar a un posible candidato de aquel que no lo es. Dadas las condiciones en las que se toma la captura de la imagen, el área del insecto puede ayudar a descartar los insectos más grandes y a los más chicos. El perímetro permite diferenciar entre insectos de similar tamaño que tengan patas o antenas, como las hormigas, las cuales tienen unas patas robustas y por tanto su perímetro es mayor que el de las moscas.

Existe la posibilidad de que el campo de visión de la cámara capture la periferia de la cámara y no sea retirada al corregir la distorsión de Barril, por lo que aparecen líneas cerca de los bordes. Para descartarlas como candidatos se mide la elongación, la cual a su vez puede dar una idea de la proporción de los insectos y permite identificar insectos alargados o sin alas extendidas.

Tras la elección de características se procede con el entrenamiento de la red neuronal (Fig. 4.15.). Esta consiste en una red con 12 capas, en la primera se introducen los parámetros que componen el vector de características (Área, Elongación y Perímetro), las 10 capas intermedias irán calculando la posibilidad que tiene el insecto de ser o no un candidato a mosca y la última capa tiene 2 salidas: Candidato/Otro.

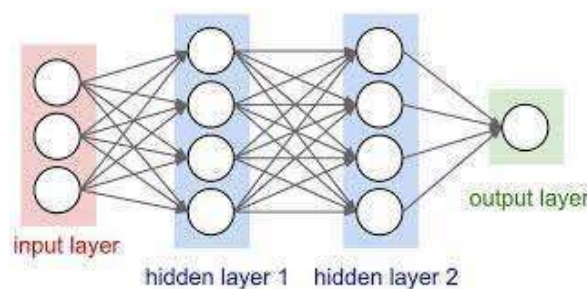


Figura 4.53. Arquitectura de una Red neuronal

Matlab cuenta con la herramienta “Neural Network Pattern Recognition app”, la cual entrena una red neuronal a partir de 2 vectores, uno con los valores de entrada (Área, Elongación y Perímetro) y otro con los valores de salida (Candidato y Otros). También es capaz de evaluar el rendimiento de la red una vez entrenada, lo cual ayuda a mejorar las tasas de acierto.

Para crear los Vectores Input y Output necesarios para la red, se programa un algoritmo en guide (Fig. 4.54.) (Diagrama 4.2.) en el cual el usuario realiza una clasificación inicial de 300 ejemplares. Dada la simplicidad de este paso no es recomendable ser muy preciso tratando de separar las moscas de insectos similares, pues esto crea confusión a la hora de determinar cuáles son verdaderos candidatos a mosca.

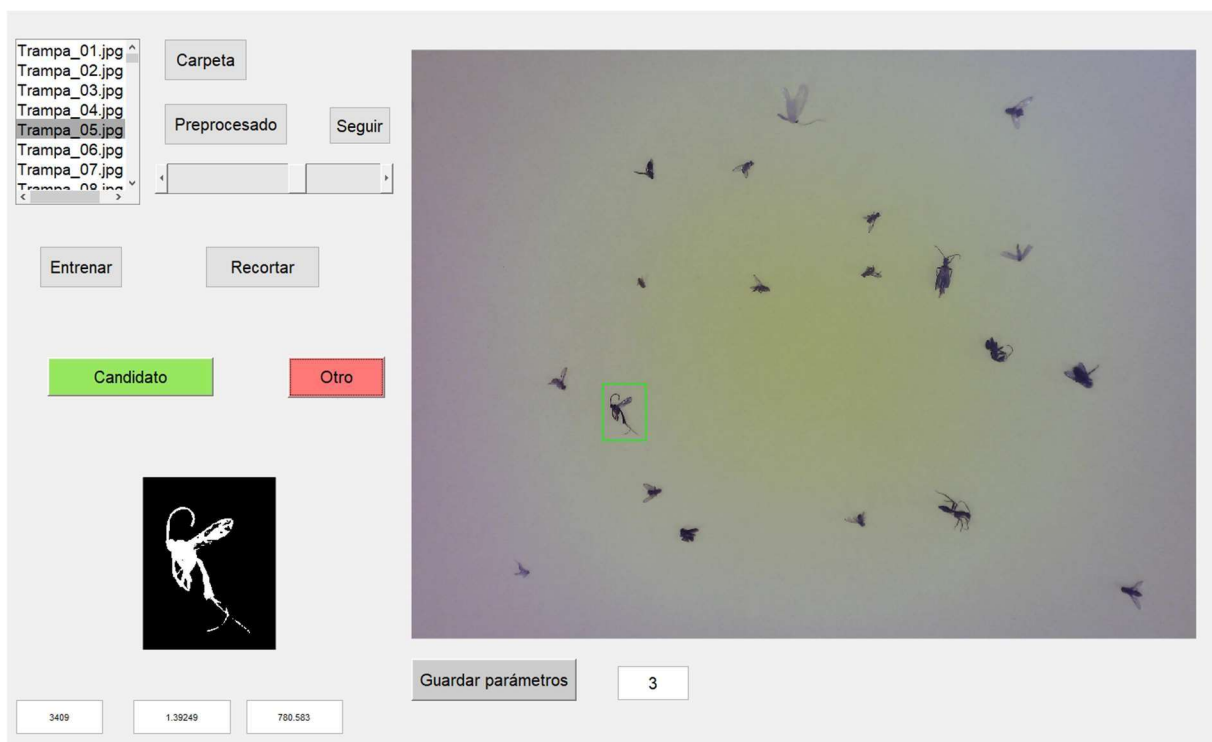


Figura 4.54. Guide Entrenamiento

Con el fin de facilitar la clasificación de los insectos, se añaden 3 casillas bajo la imagen del insecto umbralizado. Estas permiten ver los valores de Área, elongación y Perímetro medidos con el fin de entender la verdadera magnitud de lo que se está viendo.

Tras un el estudio de las 300 muestras se aprecia que las moscas tienen unos valores de área de entre 4100-1100, una elongación de entre 1-2,2 y un perímetro de entre 481-150. Por lo que, si un insecto tiene sus 3 características dentro de estos rangos, debe ser considerado un candidato.

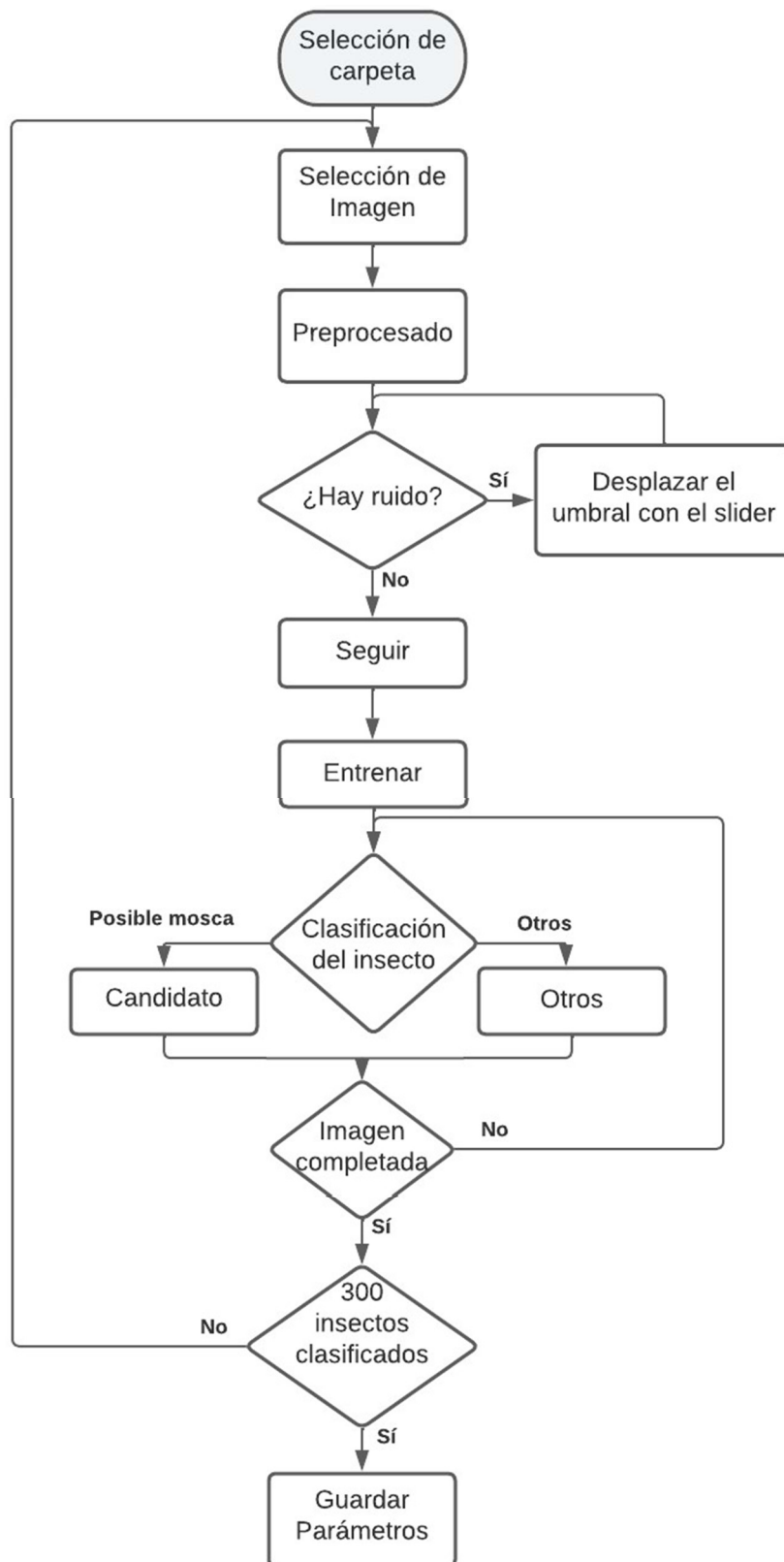


Diagrama 4.2. Entrenamiento

Tras clasificar los ejemplares se introducen los vectores guardados en la Herramienta “Neural Network Pattern Recognition app” y se entrena la Red dedicando el 70% de las muestras a entrenar, el 15% a validar el entrenamiento y el otro 15% a realizar un Test. Consultando la matriz de confusión (Fig. 4.55.) se aprecia una tasa de aciertos total superior al 95% en cuanto a la detección de posibles candidatos.



Figura 4.55. Matriz de confusión de la geometría

Implementar esta Red neuronal como algoritmo de detección es muy sencillo una vez está entrenada (Diagrama 4.3.). Tras ejecutar el análisis se obtienen todos los posibles candidatos (Fig. 4.56.), entre los que hay *Bactrocera Oleae* e insectos similares, por lo que se guardan para proseguir el análisis a continuación.

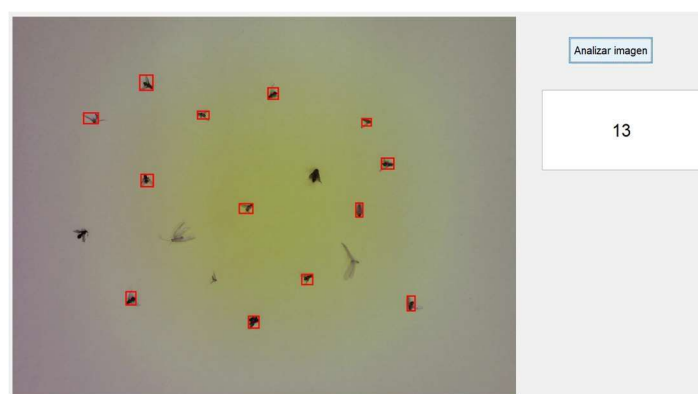


Figura 4.56. Posibles candidatos

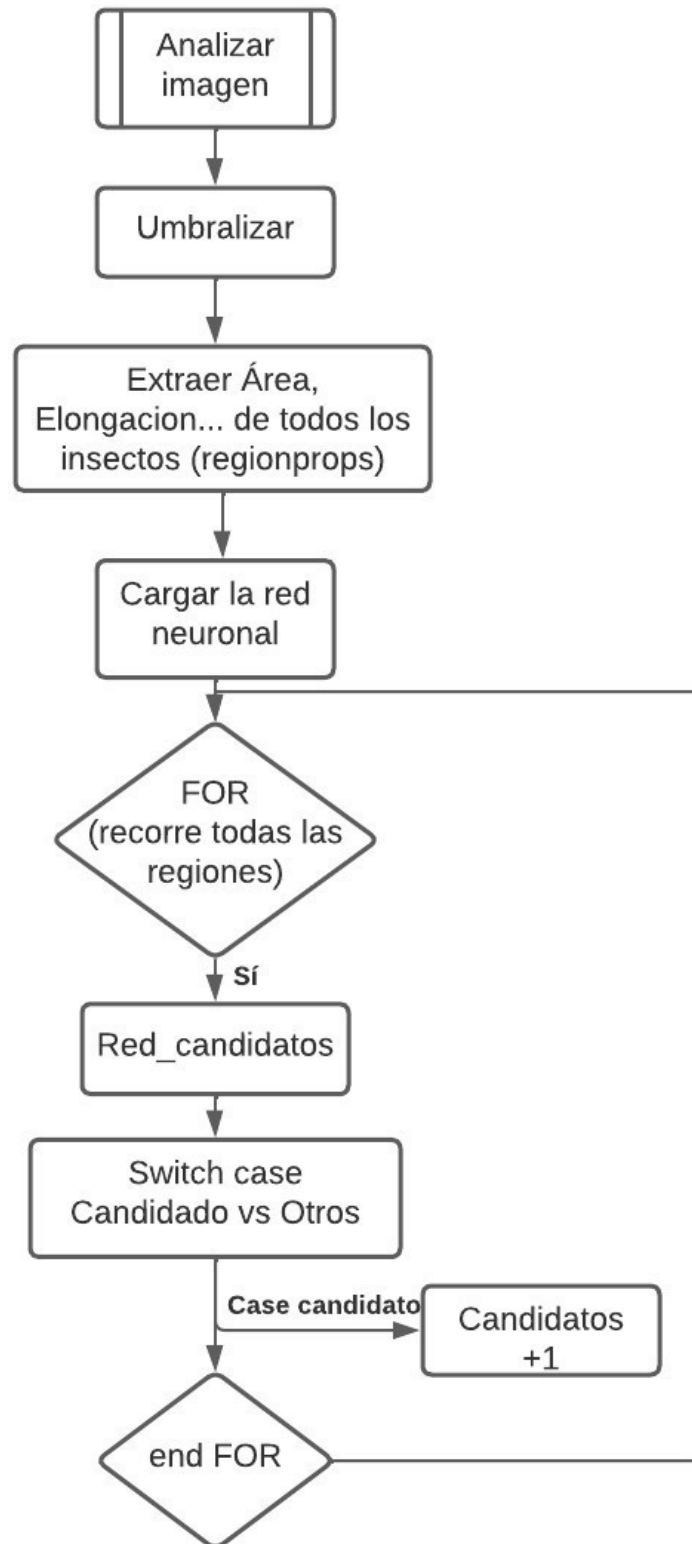


Diagrama 4.3. Análisis de la imagen

Tras la obtención de los candidatos se procede con un segundo filtrado con el objetivo de eliminar aquellos candidatos que no se identifiquen como *Bactrocera Oleae*. Este paso es más complejo debido a la morfología que presentan los insectos. Uno de los parámetros de interés en este punto es el color, ya que permite diferenciar a las moscas de aquellos insectos más claros, como pulgones o crisópidos.

El color del insecto cambia ligeramente según el punto del insecto que se analice, por lo que se aplica un filtro de la media para obtener un color más uniforme. Tras esto se extrae el color del pixel situado en el centroide del insecto y se le resta el valor correspondiente al color del fondo, obteniendo así un valor correspondiente al contraste entre el insecto y el fondo, de esta manera, si la foto se toma un día soleado o nublado no cambiará mucho.

Se hace un estudio de esta característica entre 100 insectos candidatos y se entrena una red neuronal con estos datos, eligiendo como clase 1 los insectos oscuros (moscas de varios tipos) y clase 2 aquellos insectos de color claro (pulgones y crisópidos). Gracias al alto poder discriminador de estos datos los resultados del entrenamiento muestran una eficiencia del 100% (Fig. 4.57.).

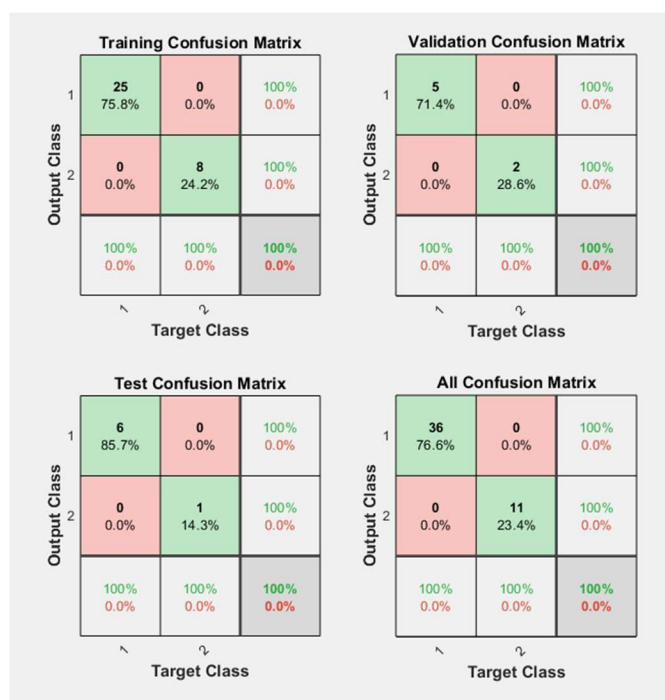


Figura 4.57. Matriz de confusión de color

Al implementar esta red en el algoritmo de Análisis se consiguen descartar candidatos, aunque algunas moscas, debido al color perdido a causa del etanol, también son descartadas si se encuentran con poca luz, ya que no pueden resaltar sus tonalidades marrones.

En este punto casi todos los candidatos son moscas del olivo (Fig. 4.58.), sin embargo, para diferenciar las moscas de los candidatos restantes deben consultarse características específicas de la especie *Bactrocera Oleae*, como vienen a ser las manchas amarillentas del tórax o las puntas negras de las alas.



Figura 4.58. Resultado del Análisis

Este tipo de análisis requieren de una alta potencia de cómputo, ya que deben analizarse cada uno de los píxeles del insecto buscando esas manchas amarillentas, lo cual ralentiza enormemente el tiempo de análisis y no garantiza el éxito, ya que la mosca puede caer de cualquier manera.

5. Resultados

El sistema de detección funciona correctamente (Fig. 4.59.), la batería mantiene una carga superior al 80% al acabar su ciclo diario y se reciben 3 imágenes de las trampas al día en el siguiente correo.

Correo: detectamoscas@gmail.com Pass: moscaolivo



Figura 4.59. Sistema autónomo de detección de plagas

Con el fin de averiguar la fiabilidad del conteo de las moscas atrapadas se prueba el software con 15 imágenes aleatorias (Tabla 4.2.) y se obtienen los siguientes resultados:

Nº Trampa	Positivos	Falsos Positivos	Falsos Negativos
4	12	3	1
13	7	2	3
32	12	3	0
37	10	0	0
44	13	1	0
55	13	1	0
63	14	4	0
68	11	1	0
74	12	3	0
79	10	2	1
82	8	2	0
87	8	1	0
93	7	3	0
17	8	3	3
23	8	2	1
TOTAL	153	31	9

Tabla 4.2. Test de detección de moscas

Tras el estudio se concluye que la mosca es detectada con un 92,6% de precisión, mientras que hay un 20,3% de los positivos no corresponden con la mosca del olivo, lo cual implica una precisión general del 75,3%

6. Presupuesto

El presupuesto se divide en dos partes, una basada en la estimación de los costes indirectos del proyecto (diseño, programación e implementación) y otra enfocada en el coste de los materiales necesarios para la construcción del sistema.

Costes indirectos:

<i>Denominación</i>	<i>Horas</i>	<i>Precio/hora</i>	<i>Total</i>
<i>Ingeniero (Diseño de hardware)</i>	30	30 €	900 €
<i>Ingeniero (Diseño de software)</i>	40	30 €	1200 €
<i>ingeniero (implementación)</i>	5	30 €	150 €
<i>Ordenador</i>		46 €	46 €
TOTAL			2296€

Costes directos:

<i>Denominación</i>	<i>cantidad</i>	<i>precio unitario</i>	<i>total</i>
<i>Raspberry</i>	1	54,82 €	54,82 €
<i>PiWatcher</i>	1	6,03 €	6,03 €
<i>Cámara ELP</i>	1	79,8 €	79,8 €
<i>Protección cámara 3D</i>	1	8 €	8 €
<i>Panel solar 15W</i>	1	25,58 €	25,58 €
<i>Regulador de carga 12V 30A</i>	1	48,28 €	48,28 €
<i>Batería</i>	1	46 €	46 €
<i>Trampas cromotrópica</i>	50	0,32 €	16 €
<i>Perfiles aluminio 30x30</i>	10	11,5 €	115 €
<i>Bolsa 100 tornillos M8</i>	1	12,74 €	12,74 €
TOTAL			412,25 €

Una vez analizados los costes y el precio de implementación del sistema automático de detección, se puede concluir que el precio total del proyecto asciende a la expresada cantidad de " DOS MIL SETECIENTOS OCHO EUROS CON VENTICINCO CÉNTIMOS" (2708,25€).

7. Conclusiones y trabajos futuros

El sistema autónomo está diseñado de tal manera que permite la implementación de nuevos dispositivos como una segunda cámara para aumentar ligeramente la resolución de la imagen.

El algoritmo de detección, a pesar de tener buenos resultados, podría tener mejores resultados en el caso de implementar correctamente una red neuronal convolucional, ya que es la solución más utilizada por otras personas y se obtienen tasas de acierto generales cercanas al 92% [12]. Este tipo de algoritmos requieren una gran potencia de cómputo y algunos conocimientos en cuanto a la modificación de redes convolucionales.

La fase de entrenamiento, al realizarse con especímenes capturados previamente, no es completamente fiel respecto a las características identificativas de cada insecto, especialmente respecto al color, por lo que puede ser necesario hacer un reentrenamiento con los especímenes que se capturen a lo largo de los meses.

7.1. Posible diseño futuro

Este sistema tiene un gran potencial para emprendedores, si se consigue simplificar los componentes para compactarlo todo podría ser una gran aportación al mundo agrícola.

Usando una placa solar de más potencia e implementando un Real Time Clock para controlar los consumos de la Raspberry y del router wifi podría reducirse considerablemente el tamaño de la batería necesitada, consiguiendo así un diseño más liviano y versátil, similares a los que ofrecen empresas como SEMIOS, sería necesario modificar el circuito de potencia del router wifi para que no fuese necesario pulsar el botón para encenderlo.

Bibliografía

[1] <<Plagas y enfermedades del olivar>> [En línea]. Available:

https://www.infoagro.com/olivo/plagas_enfermedades_olivo.htm

[2] <<Red de Alerta e Información Fitosanitaria>> [En línea] Available:

<https://www.juntadeandalucia.es/agriculturapescayderosarrollorural/raif/manuales-de-campo>

[3] <<InfoAgro _ Seguimiento de la mosca del olivo>> [En línea] Available:

https://www.infoagro.com/noticias/2017/inicio_del_seguimiento_de_la_mosca_del_olivo.asp

[4] <<BASF Agricultura _ Tratamientos>> [En línea] Available:

[https://www.agro.basf.es/es/Camposcopio/Secciones/Enfermedades-y-plagas/Mosca-del-olivo-y-tratamiento-a-parcheo/#:~:text=La%20mosca%20del%20olivo%20\(Bactrocera,de%20la%20ca%C3%ADDa%20del%20fruto.](https://www.agro.basf.es/es/Camposcopio/Secciones/Enfermedades-y-plagas/Mosca-del-olivo-y-tratamiento-a-parcheo/#:~:text=La%20mosca%20del%20olivo%20(Bactrocera,de%20la%20ca%C3%ADDa%20del%20fruto.)

[5] <<ANIA _ Biocontrol y Big Data>> [En línea] Available:

<https://www.ainia.es/ainia-news/tecnicas-de-biocontrol-y-big-data-para-combatir-las-plagas-del-olivo/>

[6] <<Ciclo biológico de la mosca del olivo>> [Vídeo] Available:

https://www.youtube.com/watch?v=py1YY_vzCRM

[7] <<Xylella fastidiosa>> [En línea] Available:

<https://www.mapa.gob.es/es/agricultura/temas/sanidad-vegetal/organismos-nocivos/xylella-fastidiosa/>

[8] <<Proyecto_FruitFlyNet>> [En línea] Available:

<https://www.phytoma.com/noticias/noticias-de-actualidad/la-uco-perfecciona-una-trampa-automatica-para-luchar-contrala-mosca-del-olivo>

[9] <<Tutorial Raspbian>> [En línea] Available:

<https://www.youtube.com/watch?v=ra6kNSIB1uA>

[10] <<Gmail con python>> [En línea] Available:

<https://www.youtube.com/watch?v=RUbTKA47Aol>

[11] <<Cromtab>> [En línea] Available:

<https://www.youtube.com/watch?v=nZes93pLxk4>

[12] <<Clasificación automática de insectos mediante CNN, Capítulo 4>> [En línea]

Available:

[https://tesis.pucp.edu.pe/repositorio/bitstream/handle/20.500.12404/20520/CORDOVA PEREZ CLAUDIA APLICACION APRENDIZAJE PROFUNDO.pdf?sequence=1&isAllowed=y](https://tesis.pucp.edu.pe/repositorio/bitstream/handle/20.500.12404/20520/CORDOVA_PEREZ_CLAUDIA_APLICACION_APRENDIZAJE_PROFUNDO.pdf?sequence=1&isAllowed=y)