

ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA



PROYECTO FIN DE CARRERA

ESTUDIO DE LAS TÉCNICAS DE PREPROCESADO DE IMÁGENES PARA LA CLASIFICACIÓN DE ESPECIES DE ARAÑAS

Autor: David Batista Plaza

Tutores: Dr. Carlos Manuel Travieso González

Dr. Jesús Bernardino Alonso Hernández

Fecha: Julio 2017

ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA



PROYECTO FIN DE CARRERA

ESTUDIO DE LAS TÉCNICAS DE PREPROCESADO DE IMÁGENES PARA LA CLASIFICACIÓN DE ESPECIES DE ARAÑAS

HOJA DE FIRMAS

Alumno

Fdo.:

Tutor

Fdo.:

Tutor

Fdo.:

Fecha: Julio 2017

ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA



PROYECTO FIN DE CARRERA

ESTUDIO DE LAS TÉCNICAS DE PREPROCESADO DE IMÁGENES PARA LA CLASIFICACIÓN DE ESPECIES DE ARAÑAS

HOJA DE EVALUACIÓN

Calificación: _____

Presidente

Fdo.:

Vocal

Fdo.:

Secretario/a

Fdo.:

Fecha: Julio 2017

Índice

Capítulo 1: Introducción	9
1.1 Introducción	9
1.2 Estado del arte.....	10
1.3 Objetivo del proyecto	10
1.4 Estructura de la memoria.....	11
1.5 Resumen	12
Capítulo 2: Base de datos y preprocesado	13
2.1 Introducción	13
2.2 Estructura de la base de datos	13
2.3 Selección de la telaraña.....	16
2.4 Filtrado espacial.....	17
2.4.1 Filtro paso bajo.....	18
2.4.2 Filtro “unsharp”	19
2.4.3 Filtro de enfatización de borde	20
2.5 Binarización y segmentación	21
2.5.1 Método de Otsu para el cálculo del umbral	22
2.5.2 Método de detección de bordes de Canny	22
2.6 Eliminación del ruido.....	23
2.6.1 Filtrado de Wiener	24
2.6.2 Operaciones morfológicas.....	24
2.7 Base de datos preprocesada	26
Capítulo 3: Extracción de características	29
3.1 Introducción	29
3.2 Análisis de componentes principales	29
3.2.1 Introducción.....	29
3.2.1 Cálculo de los componentes principales	30
3.2.1 Proceso de extracción de factores	31
3.3 Vectores comunes discriminantes	34
3.3.1 Introducción.....	34
3.3.2 Cálculo de los DCV	34
3.4 Transformada discreta del coseno.....	37
3.4.1 Introducción.....	37

3.4.1 Cálculo de la DCT	38
3.5 Resumen	38
Capítulo 4: Clasificación.....	41
4.1 Introducción	41
4.2 Máquinas de vectores soporte	41
4.2.1 Modos de funcionamiento de SVM.....	43
4.3 Uso de SVM en tareas de identificación	44
4.4 Uso de SVM en tareas de verificación.....	44
4.5 Módulos de fusión.....	46
4.6 Resumen	48
Capítulo 5: Metodología experimental.....	49
5.1 Introducción	49
5.2 Estructura del sistema	49
5.2.1 Etapa de preprocesado	50
5.2.2 Validación de datos	53
5.2.3 Extracción de características	55
5.2.4 Clasificación utilizando SVMs	57
5.3 Resumen	60
Capítulo 6: Resultados experimentales.....	63
6.1 Introducción	63
6.2 Simulaciones realizadas	63
6.3 Resultados obtenidos en identificación.....	64
6.3.1 Tamaño y número de coeficientes para la telaraña completa	64
6.3.2 Tamaño y número de coeficientes para el centro de la tela	65
6.3.3 Resultados obtenidos por los sistemas individuales	67
6.3.4 Resultados obtenidos con la fusión de sistemas.....	71
6.4 Resultados obtenidos en verificación	72
6.4.1 Resultados obtenidos por los sistemas individuales	72
6.4.2 Resultados obtenidos con la fusión de sistemas.....	73
6.4 Resumen	74
Capítulo 7: Conclusiones	77
7.1 Introducción	77
7.2 Conclusiones principales	77
7.3 Líneas futuras de trabajo	78
7.4 Resumen	79
Bibliografía	81
PLIEGO DE CONDICIONES.....	85

PRESUPUESTO	89
P.1 Desglose del Presupuesto.....	93
P.2 Recursos Materiales.....	93
P.2.1 Recursos software	94
P.2.2 Recursos Hardware	94
P.3 Trabajo Tarifado por Tiempo Empleado	95
P.4 Costes de redacción del proyecto	97
P.5 Material fungible	98
P.6 Derechos de visado del COIT	98
P.7 Gastos de tramitación y envío	99
P.8 Aplicación de impuestos.....	99
ANEXO A	101
A.1 Introducción.....	103
A.2 Descripción del contenido	103
ANEXO B	105
B.1 Descripción del contenido	107

Índice de figuras

Fig. 1: Diagrama de bloques de la etapa de preprocesado	13
Fig. 2: Ejemplo de imagen de telaraña de Eustala illicita	14
Fig. 3: Ejemplo de imagen de telaraña Cyclosa caroli.....	15
Fig. 4: Ejemplo de imagen de telaraña Nephila edulis	15
Fig. 5: Ejemplo de imagen de telaraña Araneus diadematus	16
Fig. 6: Muestra de la especie Nephila edulis	17
Fig. 7: Máscara de procesamiento espacial con $P=5$	19
Fig. 8: Aplicación del filtrado espacial "average"	19
Fig. 9: Aplicación del filtrado espacial "unsharp"	20
Fig. 10: Aplicación del filtrado espacial "prewitt"	21
Fig. 11: Segmentación mediante umbralización de Otsu	22
Fig. 12: Segmentación mediante detección de bordes de Canny	23
Fig. 13: Muestra de imágenes preprocesadas	26
Fig. 14: Selección del centro de la telaraña	27
Fig. 15: Hiperplanos de separación en un espacio bidimensional.....	42
Fig. 16: Margen de un hiperplano de separación	43
Fig. 17: Curvas ROC según el valor AUC	46
Fig. 18: Módulo de fusión "Adding scores"	47
Fig. 19: Módulo de fusión "más votado"	47
Fig. 20: Diagrama de bloques del sistema experimental	50
Fig. 21: Imágenes umbralizadas de forma automática	51
Fig. 22: Imágenes procesadas con ajuste de umbralización	51
Fig. 23: Imágenes preprocesadas por el método de Canny	52
Fig. 24: Validación Hold-Out.....	54
Fig. 25: Validación k-iteraciones.....	55
Fig. 26: Diagrama de bloques del clasificador SVM para entrenamiento	58
Fig. 27: Diagrama de bloques del clasificador SVM para test	59
Fig. 28: Diagrama de bloques del clasificador para la verificación	59
Fig. 29: Evolución del número de componentes principales de PCA	67
Fig. 30: Evolución del número de coeficientes del sistema DCT.....	70
Fig. 31: Curva ROC para el sistema PCA	72

Índice de tablas

Tabla 1: Descripción de la base de datos	14
Tabla 2: Tamaños de redimensionado de imágenes y de espacio de imagen .	52
Tabla 3: Distribución de las muestras en conjuntos de entrenamiento y test...	53
Tabla 4: Resultados para el tamaño óptimo de la telaraña completa.....	65
Tabla 5: Resultados para los coeficientes de la telaraña completa.....	65
Tabla 6: Resultados para el tamaño óptimo del centro de la telaraña.....	66
Tabla 7: Resultados para los coeficientes del centro de la telaraña.....	66
Tabla 8: Resultados obtenidos con PCA.....	68
Tabla 9: Resultados obtenidos para DCV	69
Tabla 10: Resultados obtenidos en DCT.....	70
Tabla 11: Resultados de la fusión de sistemas en identificación.....	71
Tabla 12: Resultados de los sistemas individuales en verificación	73
Tabla 13: Resultados de la fusión de sistemas en verificación	74
Tabla 14: Costes de los recursos software	94
Tabla 15: Costes de las herramientas Hardware	95
Tabla 16: Factor de corrección en función del número de horas invertidas	96
Tabla 17: Costes por Tiempo Empleado	96
Tabla 18: Presupuesto de ejecución material.....	97
Tabla 19: Coste del material fungible	98
Tabla 20: Valor del presupuesto para el cálculo del visado	99
Tabla 21: Presupuesto total sin impuestos.....	99
Tabla 22: Costes Totales del Proyecto.....	100

Capítulo 1: Introducción

1.1 Introducción

Hoy en día, conservar y proteger la biodiversidad de nuestro planeta es uno de los campos de trabajo más importantes de la comunidad científica. Salvaguardar el medio ambiente es una prioridad, y para ello, el conocimiento de las especies que pueblan la Tierra es fundamental [1]. En el caso que nos ocupa, se estudiará la especie de artrópodos quelicerados del orden Araneae [2], conocido comúnmente como araña. En particular, el objeto de estudio será la telaraña.

Se parte de que los principios mediante los cuáles las arañas fabrican sus telas son diferentes a cualquier mecanismo utilizado por el ser humano para la fabricación de cualquier material artificial. Por lo que resulta esencial investigar sobre el procedimiento natural por el cual las arañas producen su tela y por qué cada especie adopta una topología de diseño diferente a la del resto [3].

En primer lugar, se puede considerar la tela de araña como una obra de ingeniería natural, optimizada para varias funciones y repartida de manera óptima para soportar el peso para el que fue diseñada. Entre sus funciones se cuentan la de atrapar presas, proteger a las arañas y soportar daños [4].

La principal característica de la tela de araña es su resistencia, aspecto en el que supera a materiales como el Kevlar, la fibra de carbono o el acero [5]. La resistencia de las telarañas se debe a dos factores:

- La capacidad de soportar peso de los hilos, que combinan una elevada tensión de rotura con una gran ductilidad. Los hilos están formados por cadenas de proteínas entrelazadas que son segregadas mediante unas glándulas que posee la araña [6].
- La arquitectura de las telas, que forma una estructura optimizada que es capaz de absorber los impactos de las presas con eficacia.

1.2 Estado del arte

En los últimos años han surgido varios trabajos que han estudiado el comportamiento de las arañas y el modo de construcción de sus telas:

En el artículo *Construction and function of the web of Tidarren sisypoides (Araneae: Theridiidae)* se detalla que la construcción de la telaraña se puede dividir en varias fases: una primera en la que se fabrica el soporte de la telaraña; una segunda en la que se rellena haciendo diversas estructuras en horizontal y una tercera en la que se elabora la maraña superior [7].

Otro artículo, *Behavioral Characters for the Higher Classification of Orbweaving Spiders*, describe el comportamiento de las arañas durante la construcción de las telarañas, incidiendo en que no es un suceso aleatorio, sino que sus patrones de ocurrencia pueden ser sistemáticos, particulares y genéricos en niveles de su especie [8].

Por lo tanto, dado que cada subespecie de araña puede presentar un patrón determinado a la hora de construir la tela, resulta interesante estudiarla como método de reconocimiento de la subespecie en cuestión.

En la publicación *Spider Recognition by Biometric Web Analysis* se propone una serie de técnicas para llevar a cabo esta tarea, basadas en análisis biométrico, como las herramientas de procesamiento de imágenes, el Análisis de Componentes Principales y las Máquinas de Vectores Soporte para diseñar un sistema que permita una correcta identificación de las especies de araña mediante su telaraña. Con este sistema, se obtuvo una tasa de acierto del 95,44%, utilizando una base de datos formada por las imágenes de las telarañas de tres especies diferentes [9].

En el Proyecto *Reconocimiento de Arañas por medio del análisis biométrico de sus telarañas* [10] se estudia un sistema de reconocimiento basado en el anterior método, con una base de datos más extensa. Este trabajo mejoró los resultados obtenidos e introduce algunas mejoras en los métodos de extracción de características.

Es de estas dos últimas reseñas bibliográficas de las cuales se ha partido para la realización del proyecto.

1.3 Objetivo del proyecto

El objetivo principal del proyecto es implementar un sistema de reconocimiento de varias especies de arañas a partir de imágenes de sus telarañas aplicando diferentes técnicas de preprocesado y clasificación.

Se dispone de una base de datos con imágenes de telarañas de cuatro especies diferentes: *Araneus diadematus*, *Cyclosa caroli*, *Nephila edulis* y *Eustala illicita*.

Las imágenes han sido cedidas por biólogos de la Facultad de Biología de la Universidad de Oxford. Dichas imágenes fueron tomadas en diferentes ambientes de trabajo y no tienen las características de contraste y calidad necesarias para el estudio de un sistema de clasificación biométrico, por lo que habrá que realizar en primer lugar un preprocesado de la imagen para aislar la telaraña y disponer de una nueva base de datos más sencilla de tratar digitalmente.

Posteriormente, deberán estudiarse las características de cada grupo de imágenes, con el fin de extraer los parámetros que permitan diferenciar una imagen de otra de una especie diferente con la mayor claridad posible. A continuación, se deberá realizar la clasificación mediante diferentes métodos biométricos. Una vez realizada la clasificación, se deben extraer los resultados y analizarlos para corroborar que se cumplen los objetivos propuestos y se mejora el sistema de procesado y clasificación.

Todos los procesos descritos serán realizados con el lenguaje y herramientas Matlab, desde el procesado de la base de datos, hasta la clasificación y verificación de resultados.

1.4 Estructura de la memoria

La memoria de este Proyecto de Fin de Carrera está formada por siete capítulos y dos anexos. A continuación, se describe el contenido de cada uno de ellos:

- **Capítulo 1. Introducción:** Se ofrece una introducción acerca del tema tratado, en el que se explican los objetivos, describiendo los antecedentes y exponiendo la estructura del sistema desarrollado.
- **Capítulo 2. Base de datos y preprocesado:** Se describen las características de la base de datos usada y las diferentes herramientas usadas para el preprocesado, finalizando con la presentación de la estructura de la base de datos ya adecuada a las siguientes fases del sistema.
- **Capítulo 3. Extracción de características:** Se presenta la descripción de las transformaciones que han sido aplicadas a las imágenes de las telarañas de la base de datos. Los métodos utilizados son: Análisis de Componentes Principales (PCA), Vectores Comunes Discriminantes (DCV) y la Transformada Discreta del Coseno (DCT).
- **Capítulo 4. Clasificación:** Se describe del tipo de clasificador usado para la identificación y verificación. En este caso el clasificador está basado en Máquinas de Vectores de Soporte (SVM).
- **Capítulo 5. Metodología experimental:** Se describe la estructura utilizada para el sistema de clasificación con la explicación detallada de las tareas que cada bloque realiza dentro del sistema completo.

- **Capítulo 6. Resultados experimentales:** Se presentan los experimentos realizados y los resultados obtenidos para los diferentes sistemas de extracción de características usados.
- **Capítulo 7. Conclusiones:** Se explican las conclusiones y las futuras líneas de trabajo.
- **Bibliografía:** Se detalla la bibliografía usada en este proyecto.
- **Pliego de condiciones:** Se presentan las herramientas utilizadas, tanto software como hardware.
- **Presupuesto:** Se detalla el presupuesto para la elaboración de este Proyecto Fin de Carrera.
- **Anexo A:** Descripción del contenido del CD que se entrega junto con esta memoria.
- **Anexo B:** Contenido del artículo “Biometric analysis for the recognition of spider species according to their webs”, que consiste en una exposición de los trabajos realizados en el presente proyecto.

1.5 Resumen

El objetivo principal del proyecto es realizar un sistema de reconocimiento de varias especies de arañas a partir de imágenes de sus telarañas aplicando diferentes técnicas de preprocesado y clasificación.

Para lograr este objetivo, se desarrollará cada una de las fases del sistema de clasificación, buscando obtener los mejores resultados posibles, mediante métodos diferentes de preprocesado y estudiando diversas técnicas de extracción de características y de clasificación.

Todos los procesos descritos serán realizados con el lenguaje y herramientas Matlab, desde el procesado de la base de datos, hasta la clasificación y verificación de resultados.

Capítulo 2: Base de datos y preprocesado

2.1 Introducción

En este capítulo se procederá a detallar las características de la base de datos usada en este proyecto de Fin de Carrera, con el fin de realizar el sistema de reconocimiento y verificación de las especies de araña a partir de su telaraña.

Dado que las imágenes de la base de datos no tienen las características de contraste y aislamiento necesario para una correcta extracción de características y clasificación mediante métodos biométricos, se hace necesario un preprocesado previo de las imágenes, que consigan aislar la tela de araña de los elementos externos a ella. Para realizar dicho preprocesado se han utilizado diversas técnicas, que se analizarán a continuación, y se expondrán detalladamente las que definitivamente se han seleccionado para la realización de este proyecto.

El esquema de bloques del preprocesado es el siguiente:

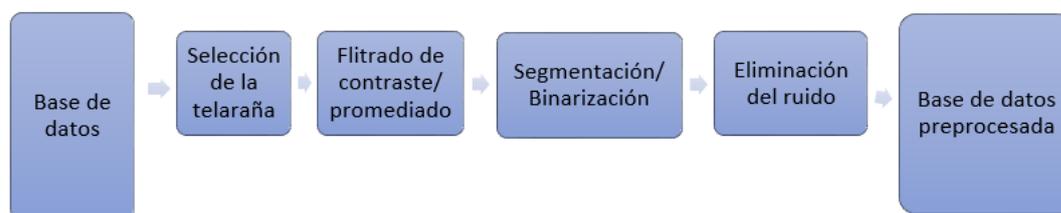


Fig. 1: Diagrama de bloques de la etapa de preprocesado

2.2 Estructura de la base de datos

La base de datos se compone de un conjunto de imágenes de telarañas pertenecientes a cuatro especies diferentes. Tres de ellas son del tipo Araneidae,

a saber: *Araneus diadematus*, *Eustala illicita* y *Cyclosa caroli*. La cuarta pertenece al tipo Nephilidae y su nombre científico es *Nephila edulis*. Todas las fotografías se tomaron en ambientes controlados. En la siguiente tabla se detalla el número de imágenes de cada especie y el tamaño de las mismas:

Especie	Número de imágenes
<i>Eustala illicita</i> (Araneidae)	76
<i>Cyclosa caroli</i> (Araneidae)	38
<i>Nephila edulis</i> (Nephilidae)	33
<i>Araneus diadematus</i> (Araneidae)	38

Tabla 1: Descripción de la base de datos

De la especie *Eustala illicita*, a pesar de que las imágenes se obtuvieron en un ambiente controlado en laboratorio y en escala de grises, no se ha cuidado especialmente la iluminación ni el contraste de la telaraña, debido a la presencia de objetos tanto claros como oscuros que hacen que la tarea de aislamiento de la telaraña sea bastante complicada. El tamaño de las imágenes facilitadas es 3872x2592 píxeles.



Fig. 2: Ejemplo de imagen de telaraña de *Eustala illicita*

Las imágenes de la especie *Cyclosa caroli* han sido tomadas también en escala de grises, pero con una mejor iluminación y mejor aislamiento, pero la calidad es algo menor. El tamaño de las imágenes no es el mismo para todo el conjunto, ya que 13 de ellas son de 2560x1920 y el resto de 3872x2592 píxeles.



Fig. 3: Ejemplo de imagen de telaraña *Cyclosa caroli*

La tercera especie, *Nephila edulis* posee las imágenes más cuidadas en cuanto a calidad y aislamiento de la tela de araña en el laboratorio, por lo que la mayoría de ellas son las más aceptables para realizar un preprocesado óptimo. El tamaño es de 3872x2592 píxeles.



Fig. 4: Ejemplo de imagen de telaraña *Nephila edulis*

La cuarta y última, *Araneus diadematus*, presenta imágenes con características similares a las de la especie *Nephila edulis*. El fondo oscuro y uniforme facilita que la tela de araña tenga un mejor contraste, óptimo también para un buen preprocesado. No obstante, la calidad en un 30% (aproximadamente) de las imágenes no es del todo buena y presentan falta de nitidez. El tamaño también es de 3872x2592 píxeles.

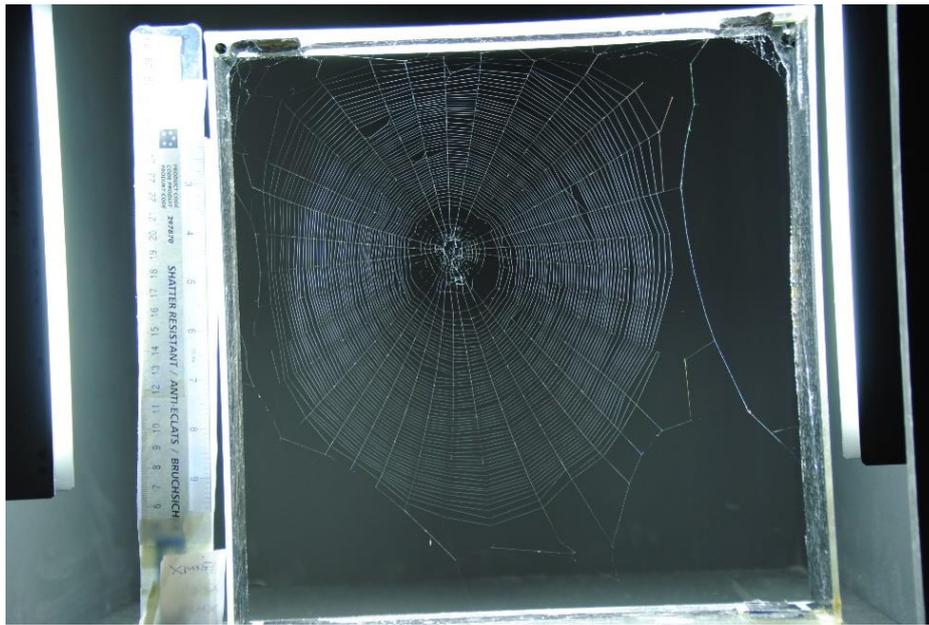


Fig. 5: Ejemplo de imagen de telaraña *Araneus diadematus*

Se han ido presentando durante este capítulo muestras de las imágenes de la telaraña de cada especie. La totalidad de las mismas se incluye en el CD adjunto a esta memoria.

2.3 Selección de la telaraña

Las imágenes de las que consta la base de datos no contienen la telaraña aislada, sino que es una fotografía tomada en un entorno controlado en el cuál hay presentes más elementos que la propia telaraña. Por ello, para realizar un correcto preprocesado de las imágenes, se hace necesario recortar y redimensionar las mismas para quedarnos solo con la telaraña, que es lo único que nos interesa para este proyecto.

La selección de la telaraña no es un proceso completamente automático, sino que se debe seleccionar manualmente qué parte nos interesa de la imagen. No obstante, esto se hace mediante un script de Matlab que consta de los siguientes pasos:

- Mediante la función *imcrop* de la *Image Processing Toolbox* de Matlab [11], se crea una herramienta que permite seleccionar el área de la telaraña, y al hacer doble clic esta se guarda en el workspace de Matlab.
- Una vez seleccionada el área, se realiza una cuadratura de la misma de forma automática, mediante un código que iguala el ancho y el alto de la imagen haciendo que la mayor de las dos dimensiones se iguale a la menor. De esta manera no se pierde la relación de aspecto al redimensionarla.

- Se guarda la imagen resultante en carpetas de destino contenidas dentro de las carpetas de cada conjunto de imágenes por especie de araña.

Como ejemplo resultante de este paso, se presenta la imagen original y la imagen recortada a continuación, utilizando una muestra de la especie *Nephila edulis*:

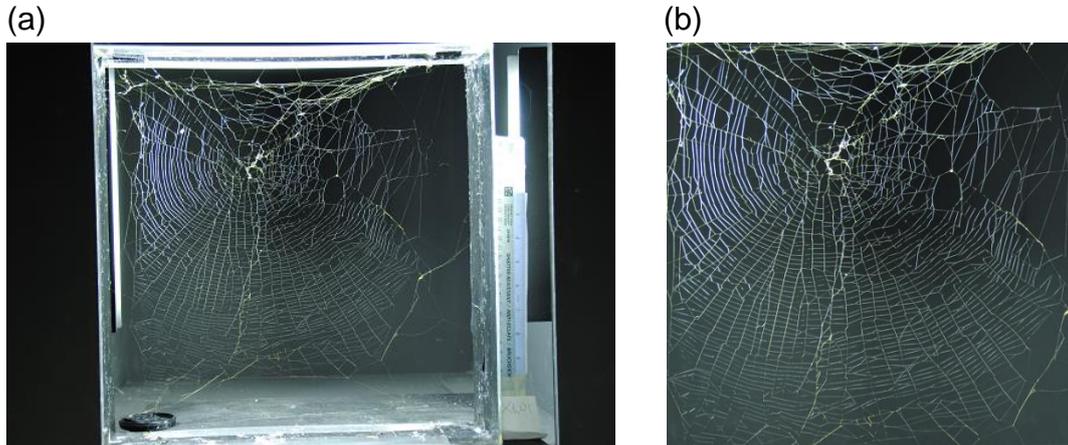


Fig. 6: Muestra de la especie *Nephila edulis* (a) imagen original (b) imagen recortada

Se observa como en efecto, una vez seleccionada el área de interés de manera manual, el programa realiza la cuadratura quedándonos con una imagen de menor tamaño que la original, y apta para ser preprocesada.

2.4 Filtrado espacial

Con el fin de conseguir mejoras en el contorno de los hilos de la telaraña y ser capaces de diferenciar la telaraña del fondo de la imagen se van a aplicar diferentes técnicas para conseguir un buen realce o un incremento del contraste, que consiste en resaltar el cambio de luminosidad entre las zonas más oscuras de la imagen y aquellas que son más claras [10], consiguiendo mejoras en la claridad y el enfoque.

En primer lugar, se realiza una conversión de la imagen original a escala de grises mediante la función `rgb2gray` de Matlab. Una vez hecho esto, se realiza un filtrado espacial [12] a la imagen con la función `imfilter`, con tres tipos de filtros cuyo resultado se analiza para seleccionar finalmente el que mejor resultado presente.

El filtrado espacial es la operación que se aplica a una imagen o conjunto de imágenes con el fin de resaltar detalles espaciales de manera que se mejore el resultado final visual o se facilite el procesado posterior [10]. Las operaciones espaciales se basan en un procesado con la siguiente forma:

$$v = T(u, u_1, u_2, \dots, u_{P^2-1}) \quad (2.1)$$

donde u y v son, respectivamente, los valores original y transformado de un cierto pixel, y $(u_1, u_2, \dots, u_{P^2-1})$ son los valores de intensidad de los pixels adyacentes al que se estudia. La función T , común a todos los pixels de la imagen, se toma en las operaciones espaciales de tipo convolutivo como una función lineal que representa los valores de intensidad de entrada, es decir,

$$v = \lambda u, + \sum_{i=1}^{P^2-1} \lambda_i u_i \quad (2.2)$$

El tipo de operación espacial aplicada dependerá de los coeficientes elegidos $(\lambda, \lambda_1, \dots, \lambda_{P^2-1})$. Por tanto, dado que los coeficientes definen enteramente la operación, se usa una máscara de valores para resumir estos operadores. Esta máscara consiste en una rejilla de $P \times P$ píxeles, con P impar y centrada en torno al pixel analizado, en la que cada casilla contiene el coeficiente que multiplicará a cada uno de los píxeles que intervenga en la operación. De esta manera, se puede interpretar con rapidez en qué consiste el operador con solo analizar la máscara.

Como ya se ha explicado antes, se realizará el filtrado espacial con tres tipos de filtros, que se implementarán mediante la función *fspecial*. Estos son:

- Filtro paso bajo de promediado “average”
- Filtro de mejora del contraste “unsharp”
- Filtro de enfatización de borde “prewitt”.

2.4.1 Filtro paso bajo

En este tipo de filtro [12], los coeficientes de la máscara (se presenta un ejemplo en la Fig.7) serán todos de signo positivo. Las diferentes implementaciones resaltarán más o menos el píxel central con respecto a los vecinos, pero el comportamiento del filtro es similar. Habitualmente los coeficientes se normalicen para que su suma sea unitaria. De este modo, la ganancia del filtro para frecuencias espaciales nulas es igual a uno, de manera que no se altera la componente continua de la imagen.

El objetivo de este tipo de filtrado es reducir el ruido de adquisición cuando sólo se tiene una única imagen. El inconveniente de esta técnica está en el suavizado de las transiciones (por el recorte de aquellas componentes espectrales de alta frecuencia) provocando que el aspecto final de la imagen presente un cierto desenfoque, directamente proporcional al tamaño del filtro.

λ_1	λ_2	λ_3	λ_4	λ_5
λ_6	λ_7	λ_8	λ_9	λ_{10}
λ_{11}	λ_{12}	λ	λ_{13}	λ_{14}
λ_{15}	λ_{16}	λ_{17}	λ_{18}	λ_{19}
λ_{20}	λ_{21}	λ_{22}	λ_{23}	λ_{24}

Fig. 7: Máscara de procesamiento espacial con $P=5$

En la siguiente figura se puede observar la imagen resultante de aplicar un filtrado espacial a la imagen en escala de grises mediante el filtro “average” del paquete *Image Processing Toolbox* de Matlab.

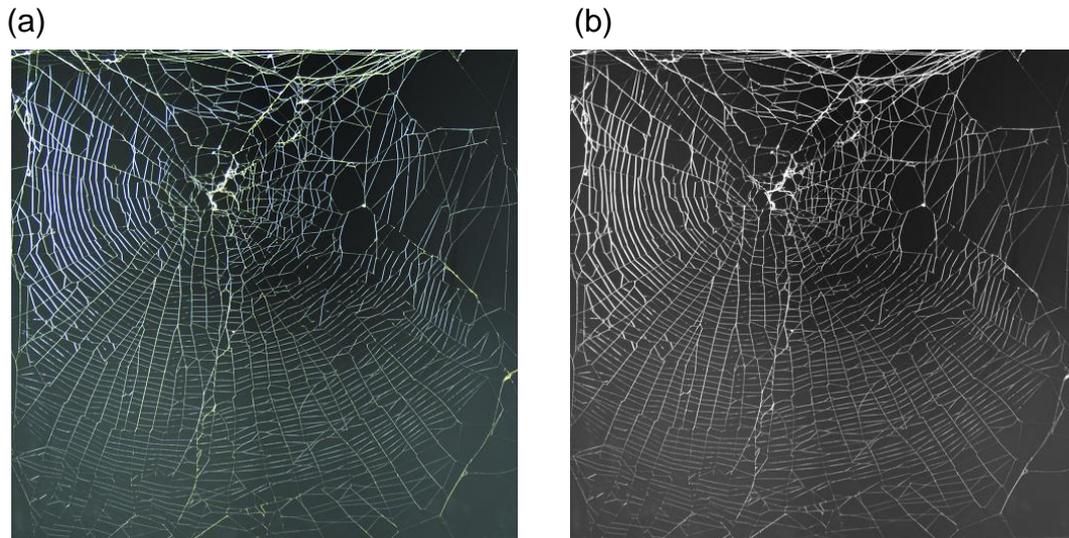


Fig. 8: Aplicación del filtrado espacial “average”, (a) imagen original recortada, (b) imagen tras filtrado

2.4.2 Filtro “unsharp”

Los filtros paso bajo y paso alto son en cierta medida extremos debido a que enfatizan bien respectivamente solo las componentes con baja frecuencia o bien solo las componentes con alta frecuencia. Los primeros, según se ha visto ya, reducen el ruido de la imagen, pero difuminan las transiciones. Los segundos, sin embargo, enfatizan las transiciones de dicha, pero a costa de eliminar aquellas componentes que tienen intensidad homogénea.

Un filtro que mezcla ambos comportamientos es el que habitualmente se ha llama *unsharp masking* [11], dado que opera eliminando una parte de la imagen paso bajo correspondiente a la imagen original.

Si se denomina O a la imagen original, HPO se corresponde con la versión paso alto de la imagen original, LPO con su versión paso bajo, y UMO hace referencia a la imagen filtrada a través de este filtro, el proceso se explica a continuación:

$$\begin{aligned}
 O &= HPO + LPO \\
 UMO &= A * O - LPO \\
 &= A * O - (O - HPO) \\
 &= (A - 1) * O + HPO
 \end{aligned}
 \tag{2.3}$$

De esta manera, si la constante A es mayor que 2, el filtro *unsharp masking* equivaldría a añadir una parte de la componente paso alto de la imagen a la versión original amplificada de dicha imagen. Con esto, se podrá lograr enfatizar las transiciones sin peligro de perder información de la estructura original de la imagen.

En la siguiente figura se puede observar la imagen resultante de aplicar escala de grises y un filtrado espacial a la imagen original mediante el filtro “unsharp” del paquete *Image Processing Toolbox* de Matlab.

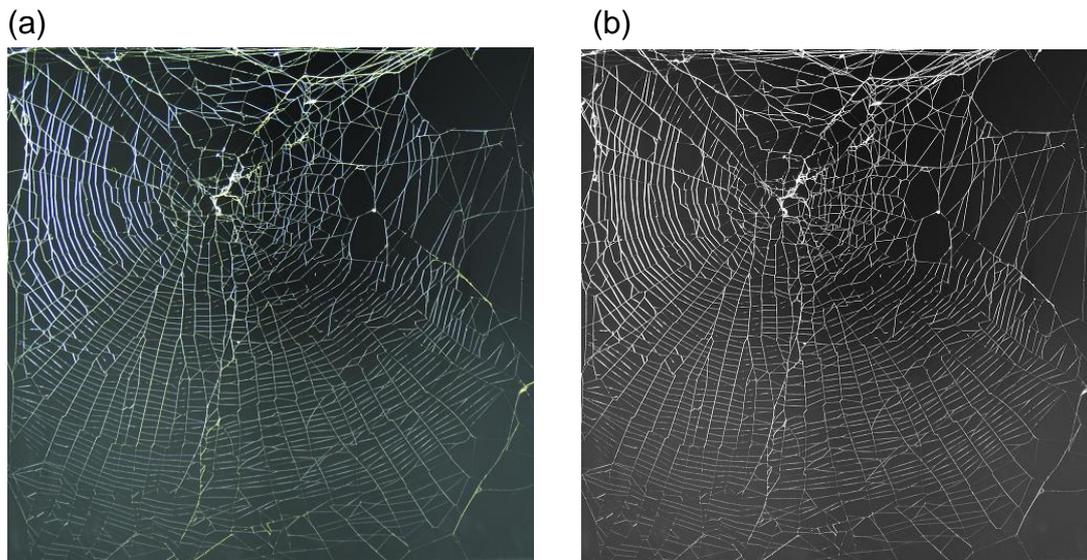


Fig. 9: Aplicación del filtrado espacial "unsharp", (a) Imagen original recortada, (b) imagen tras filtrado

2.4.3 Filtro de enfatización de borde

Bajo el nombre de filtro de enfatización de borde pueden entrar aquellos operadores que realizan operaciones de filtrado paso alto. Sin embargo, esta denominación se suele utilizar para operadores de tipo gradiente o de primera derivada, y, en general, con carácter direccional.

En la práctica, esto es, cuando se trabaja con imágenes digitalizadas, esta operación no suele realizarse en un dominio continuo, sino que se aproxima por una versión discreta, que se obtiene mediante máscaras.

En el actual proyecto, se hará uso de los operadores de Prewitt [12]. Estos son operadores de 3x3 de cálculo de derivadas vertical y horizontal, donde el operador de derivada vertical enfatiza los bordes horizontales, y el operador de derivada horizontal hace lo propio con los bordes verticales. Las máscaras de las que hace uso son, respectivamente para derivadas vertical y horizontal:

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad (2.4)$$

En la siguiente figura se puede observar la imagen resultante de aplicar escala de grises y un filtrado espacial a la imagen original mediante el filtro “prewitt” del paquete *Image Processing Toolbox* de Matlab.

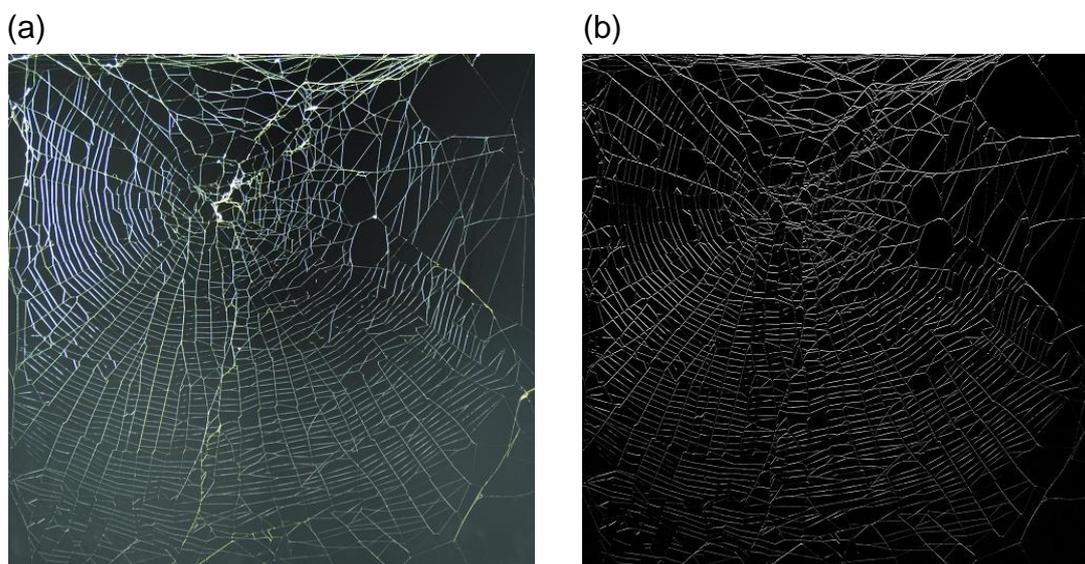


Fig. 10: Aplicación del filtrado espacial "prewitt", (a) Imagen original recortada, (b) imagen tras filtrado

2.5 Binarización y segmentación

El proceso de binarización consiste en transformar la imagen a un formato binario (blanco y negro) de forma que se conserven las propiedades fundamentales de la imagen, lo que va a permitir procesar de forma clara el contorno de los hilos de la telaraña. Este proceso se realizará experimentalmente de dos formas diferentes. Una estará basada en el método de Otsu para el cálculo del umbral y posterior binarización, y la otra mediante el método de detección de bordes de Canny.

Las imágenes que se presentarán a partir de este punto han sido las resultantes de aplicar un filtrado paso-bajo (función “average”) previo a la segmentación, dado que fue el método que mejores resultados visuales ofrecía, al mantener el contraste de la telaraña sin oscurecer partes importantes de la misma.

2.5.1 Método de Otsu para el cálculo del umbral

En primer lugar, se utilizará la función “*im2bw*” de la *Image Processing Toolbox* de Matlab. Esta función transforma imágenes a color o en escala de grises en imágenes binarias a partir de un determinado umbral, es usado como parámetro de entrada en dicha función [14]. Se puede resumir este proceso con la siguiente ecuación:

$$g(x,y) = \begin{cases} 0, & f(x,y) < \text{umbral} \\ 1, & f(x,y) \geq \text{umbral} \end{cases} \quad (2.5)$$

Este umbral se obtiene con la función “*graythresh*”, basada en el método de Otsu para el cálculo del umbral. Este método consiste, resumidamente, en calcular el cociente entre las varianzas entre dos segmentos y la varianza dentro de cada segmento y buscar un valor umbral para el que este cociente sea máximo. Este umbral en teoría será el óptimo para realizar la binarización.

Un ejemplo de la imagen binarizada obtenida, comparada con la imagen en escala de grises tras el filtrado espacial, es el siguiente:

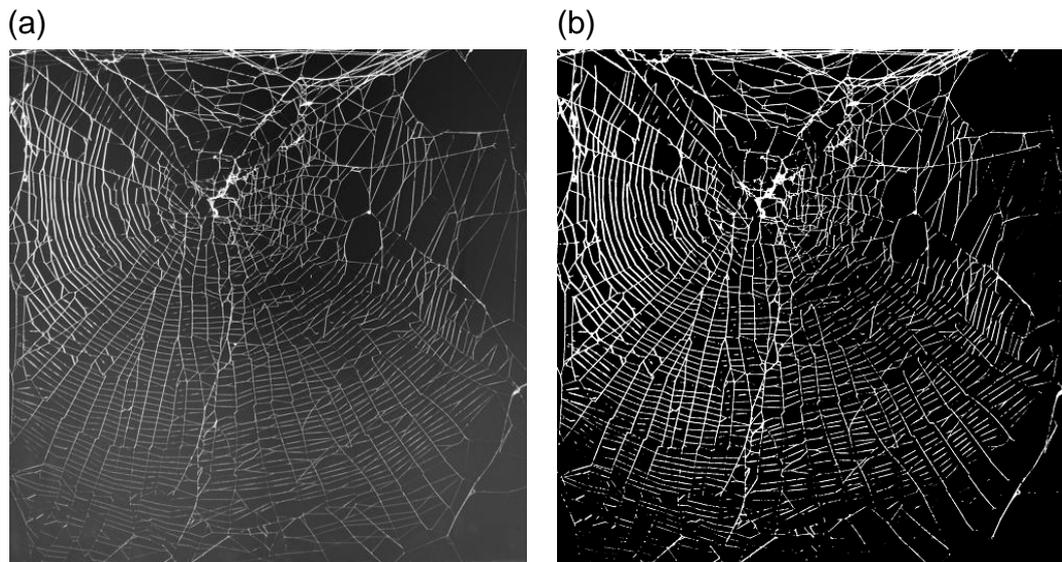


Fig. 11: Segmentación mediante umbralización de Otsu
(a) Imagen en escala de grises, (b) imagen binarizada

2.5.2 Método de detección de bordes de Canny

Como alternativa a este método, se realizará también una segmentación y binarización a partir de la detección de bordes mediante el método de Canny [15]. Este algoritmo está considerado como uno de los métodos más usados para la detección de contornos. Se puede resumir en los siguientes pasos:

1. Se convoluciona la imagen con un filtro gaussiano con el fin de suavizarla y eliminar ruidos.
2. Se obtiene el gradiente de la imagen ya suavizada, con el objeto de determinar en qué píxeles la variación es mayor, es decir, cuando mayor es el módulo del vector gradiente. Se fija también la dirección del vector gradiente.
3. La matriz M , que corresponde al módulo del gradiente de la función gaussiana, presentará valores mayores allí donde la variación de la intensidad es grande. Se eliminarán entonces los píxeles que no son máximos locales en la dirección del gradiente.
4. Se realiza una umbralización doble buscando determinar los píxeles del borde. Es decir, se marcan los píxeles con valor superior a un umbral $T1$ y se marcan aquellos píxeles vecinos a los primeros con valor superior a un segundo umbral $T2$ ($T2 < T1$). Con esto se consigue eliminar falsos bordes o bien bordes dobles.

En la práctica se utilizará la función de Matlab “*edge*”, que retorna una imagen ya binarizada tras detectar los bordes de la misma mediante el método “Canny”.

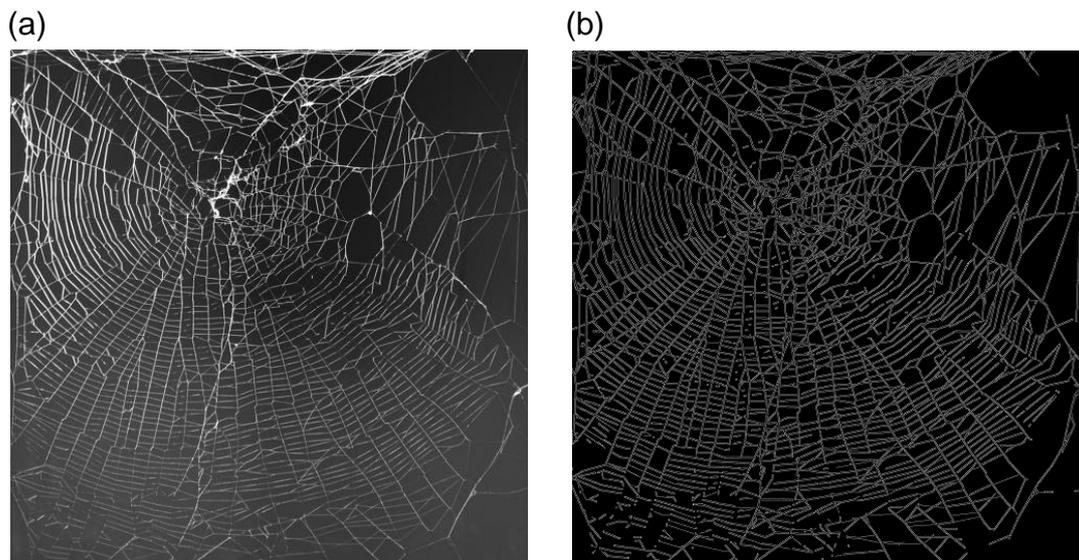


Fig. 12: Segmentación mediante detección de bordes de Canny
 (a) Imagen en escala de grises, (b) imagen binarizada

2.6 Eliminación del ruido

Una vez realizada la binarización, el siguiente paso consiste en realizar un realce de la telaraña de la imagen y eliminar el ruido de la misma en la medida de lo posible. Para ello se va a hacer uso del filtrado de Wiener, de operaciones morfológicas como clean o spur y de erosión y dilatación.

2.6.1 Filtrado de Wiener

El filtro de Wiener tiene como propósito, utilizando métodos estadísticos, reducir el ruido presente en la señal de entrada de modo que a la salida del filtro, la señal se aproxime lo más posible a una señal sin ruido.

Los filtros Wiener tienen la función de suavizar la imagen realizando cambios en las áreas donde más aparente es el ruido, pero procurando mantener aquellas áreas en las que los detalles están presentes y el ruido es menor. El filtro Wiener, por lo tanto, adapta su funcionamiento a la varianza local de la imagen [16]; allí donde la varianza sea mayor, el filtro realiza un suavizado menor y donde sea menor, realiza un mayor suavizado, consiguiendo así un mejor realce.

Este filtrado se va a implementar a partir de la función “*wiener2*” de la *Image Processing Toolbox* de Matlab. Esta función calcula la media local y la varianza alrededor de cada uno de los píxeles mediante las siguientes fórmulas:

$$\mu = \frac{1}{NM} \sum_{n_1, n_2 \in \eta} a(n_1, n_2) \quad (2.6)$$

$$\sigma^2 = \frac{1}{NM} \sum_{n_1, n_2 \in \eta} a^2(n_1, n_2) - \mu^2, \quad (2.7)$$

Donde η se corresponde con la vecindad local $N \times M$ calculada para cada píxel de la imagen a . Tras esto, la imagen se filtra usando estas estimaciones:

$$b(n_1, n_2) = \mu + \frac{\sigma^2 - v^2}{\sigma^2} (a(n_1, n_2) - \mu), \quad (2.8)$$

Siendo v^2 la varianza del ruido. Si esta no se da, la función utiliza la media calculada de todas las varianzas locales. En la práctica, se ha elegido un bloque de 2×2 , ya que fue con el que se obtuvo el mejor efecto deseado en la imagen.

2.6.2 Operaciones morfológicas

Las operaciones morfológicas [17] simplifican imágenes y conservan las principales características de forma de los objetos. Un sistema de operadores de este tipo y su composición, permiten que las formas subyacentes sean identificadas y reconstruidas de forma óptima a partir de sus formas distorsionadas y ruidosas. Mediante estas operaciones, se busca eliminar el ruido aún presente en las imágenes y también adecuarlas a una correcta extracción de características.

Para la eliminación del ruido se hará uso de la función “*bwmorph*”, con la que pueden implementarse diferentes tipos de operaciones morfológicas. Durante esta fase, las operaciones que se utilizarán serán “*clean*” y “*spur*”.

La operación “*clean*”, elimina los píxeles aislados (aquellos unos individuales que están rodeados de ceros), es decir, aquellos píxeles que se presenten de la siguiente manera:

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (2.9)$$

Por otro lado, la operación “*spur*” elimina las componentes espúreas tales como ramas o esquinas salientes, como se puede observar a continuación:

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix} \longrightarrow \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix} \quad (2.10)$$

Por otro lado, con las imágenes procesadas mediante el método de Canny, se van a realizar las operaciones de dilatación y erosión. La primera operación se realizará para rellenar el pequeño espacio en negro entre el borde superior y el inferior de cada hilo de la tela de araña, y a continuación se realizará la erosión para reducir proporcionalmente a la dilatación anterior el grosor de cada hilo.

El resultado de la operación de dilatación es el conjunto de elementos de manera que al menos un elemento del conjunto B está en el conjunto X, cuando B se desplaza sobre X:

$$\delta_B(X) = X \oplus B = \{x \mid X \cap B_x \neq \emptyset\} \quad (2.11)$$

Esta operación representa un crecimiento gradual del conjunto X. Al pasar el elemento estructurante dentro del conjunto, éste no se modifica, pero al desplazar B en la frontera del conjunto X, se produce una expansión del conjunto resultante.

Por otro lado, la transformación de la erosión es el resultado de averiguar si el elemento estructurante B está completamente incluido en el conjunto X. Si esto no sucede, el resultado de la erosión se corresponde con el conjunto vacío:

$$\varepsilon_B(X) = X \ominus B = \{x \mid B_x \subseteq X\} \quad (2.12)$$

Si los objetos de la escena son menores que el elemento estructurante, éstos tenderán a desaparecer.

Las operaciones de dilatación y erosión se implementan mediante la función “*imdilate*” y la función “*imerode*”, respectivamente. Tanto para una operación como para la otra, es necesario definir una geometría determinada al elemento estructurante, que será parámetro de entrada de esas funciones. En el caso que nos ocupa se ha optado por una estructura en disco para ambas, ya que era la que mejores resultados presentaba.

A continuación, se presentan las imágenes ya preprocesadas completamente tras aplicar las técnicas de eliminación de ruido y las operaciones morfológicas, tanto para el método de Otsu como para el de Canny.

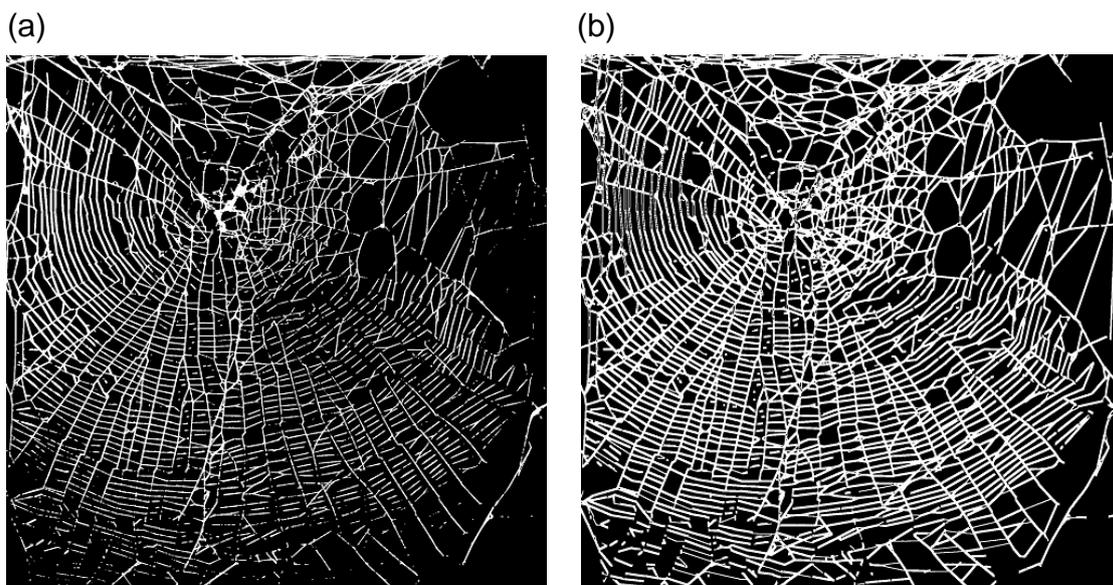


Fig. 13: Muestra de imágenes preprocesadas
(a) Método de Otsu, (b) Método de Canny

2.7 Base de datos preprocesada

Si se estudia la base de datos original de las imágenes se observa que en el centro es donde se encuentran las mayores diferencias entre una y otra especie. Por lo tanto, se puede deducir que será en esta parte donde se encuentre la mayor parte de la información que nos permita diferenciar entre una especie y otra. Resulta interesante pues construir otra base de datos aparte de la original ya preprocesada que contenga imágenes del centro de la telaraña.

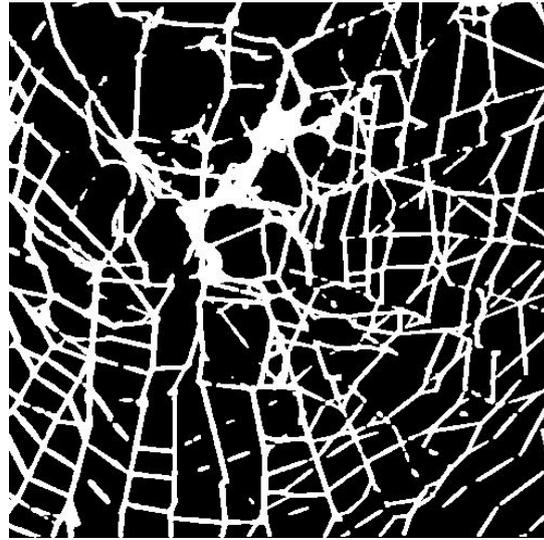
Para la confección de esta nueva base de datos, se repite el procedimiento explicado en el apartado 2.2, cambiando el área de selección, que sería ahora el centro de la telaraña. Se ubicaría la nueva base de datos junto a la anterior, diferenciándolas en carpetas diferentes. La base de original iría en una carpeta de nombre “*full*” y la nueva en otra de nombre “*center*”, en el mismo directorio.

A continuación, se muestran algunos ejemplos de imágenes de la construcción de la nueva base de datos seleccionando el centro de la telaraña:

(a)



(b)



(c)



(d)



*Fig. 14: Selección del centro de la telaraña
(a) y (c) imágenes de la tela completa
(b) y (d) imágenes del centro de la telaraña*

2.8 Resumen

A lo largo del proyecto se ha empleado una base de datos que contiene 185 imágenes y compuesta por cuatro especies diferentes, tres de la subespecie Araneidae y otra de la especie Nephilidae. Las imágenes de esta colección fueron tomadas en ambientes controlados (jaulas donde las arañas construyen sus telas), por lo que las condiciones lumínicas y ambientales varían de una

clase a otra, siendo en general poco aptas para el tratamiento tal y cómo se facilitan, por lo que deberán someterse a un preprocesado.

Realizar un preprocesado de las imágenes es fundamental para la base de datos de la que se dispone, ya que es conveniente aislar la telaraña del resto de la imagen, recortando en primer lugar la misma para tener una muestra que contenga únicamente la zona donde está la tela. Una vez hecho, se utilizarán dos técnicas de segmentación, la de umbralización de Otsu y la de detección de bordes de Canny. Se obtiene como resultado una base de datos preprocesada para cada método.

Si se estudia la colección de las imágenes de telarañas se observa que en el centro es donde más se diferencian las imágenes de una especie de las de las otras. Por lo tanto, se creará una nueva base de datos que contendrá las imágenes recortadas del centro de la telaraña, tanto para un método como para el otro.

Capítulo 3: Extracción de características

3.1 Introducción

El término “extracción de características” hace referencia al método consistente en extraer algunos parámetros de las imágenes; en este caso podrían ser por ejemplo el radio de los hilos o el área de la imagen, entre otras.

La extracción de características se inicia a partir de un conjunto inicial de datos medidos y genera valores derivados destinados a ser informativos y no redundantes. La extracción de características debe ser capaz de reducir las dimensiones de los datos y además puede mejorar la tasa de generalización de manera que los costes computacionales del clasificador sean menores.

En las siguientes secciones explicarán los diferentes métodos que se han implementado para extraer la información contenida en las imágenes. A saber: Análisis de Componentes Principales (PCA), Vectores Comunes Discriminantes (DCV) y la Transformada Discreta del Coseno (DCT).

3.2 Análisis de componentes principales

3.2.1 Introducción

Un problema importante cuando se aborda el análisis de datos multivariantes es la reducción de la dimensionalidad. Se podrá reducir la dimensión del problema si se puede describir con precisión los valores de p variables por un pequeño subconjunto $r < p$ de ellas, a costa de una pequeña pérdida de información [18].

El Análisis de Componentes Principales (PCA) es una técnica estadística utilizada que tiene como objetivo que dadas n observaciones de p variables, se analiza si se puede representar esta información con un menor número de variables que sean combinaciones lineales de las originales.

PCA es un método lineal que busca reducir la dimensionalidad siendo capaz de mantener la información intrínseca de los datos a la entrada [10]. Si las variables están incorreladas de inicio, no tendría sentido realizar entonces las técnicas de PCA.

3.2.1 Cálculo de los componentes principales

El método para el cálculo de los componentes principales es el siguiente [19]:

Considerando una serie de variables (x_1, x_2, \dots, x_p) sobre un grupo de objetos, se busca calcular, a partir de ellas, un nuevo conjunto de variables (y_1, y_2, \dots, y_p) que estén incorreladas entre sí y cuyas varianzas decrezcan de manera progresiva.

Cada y_j es una combinación lineal de las x_j originales, esto es:

$$y_j = a_{j1}x_1 + a_{j2}x_2 + \dots + a_{jp}x_p = \mathbf{a}'_j \mathbf{x} \quad (3.1)$$

donde $\mathbf{a}'_j = (a_{1j}, a_{2j}, \dots, a_{pj})$ es un vector de constantes, y

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_p \end{bmatrix} \quad (3.2)$$

Si lo que se pretende es maximizar la varianza, una forma simple consiste en aumentar los coeficientes a_{ij} . Con el fin de mantener la ortogonalidad de la transformación es necesario que el módulo del vector $\mathbf{a}'_j = (a_{1j}, a_{2j}, \dots, a_{pj})$ sea 1:

$$\mathbf{a}'_j \mathbf{a}_j = \sum_{k=1}^p a_{kj}^2 = 1 \quad (3.3)$$

El primero de los componentes se calcula eligiendo a_1 de forma que la varianza de y_1 sea lo mayor posible, siendo $\mathbf{a}'_1 \mathbf{a}_1 = 1$. El segundo componente principal se calculará obteniendo a_2 de forma que la variable obtenida a partir de esta (y_2) esté incorrelada con y_1 .

A continuación se procederá a elegir y_1, y_2, \dots, y_p , de forma que estén incorreladas entre sí, de manera que la varianza de las variables aleatorias obtenidas vaya siendo cada vez menor.

3.2.1 Proceso de extracción de factores

Como se ha explicado antes, se quiere elegir a_1 de modo que se maximice la varianza de y_1 sujeta a la restricción de que $a_1' a_1 = 1$.

$$Var(y_1) = Var(a_1' x) = a_1' \Sigma a_1 \quad (3.4)$$

Con el fin de maximizar una función multivariable sujeta a restricciones, se utiliza el método de los multiplicadores de Lagrange. Se busca en concreto maximizar la función $a_1' \Sigma a_1$ sujeta a la restricción $a_1' a_1 = 1$. Se observa que la incógnita es a_1 , que se corresponde con el vector desconocido que optimiza la combinación.

De esta forma, se construye la función L:

$$L(a_1) = a_1' \Sigma a_1 - \lambda(a_1' a_1 - 1) \quad (3.5)$$

Se busca el máximo, realizando la derivada e igualando a 0:

$$\frac{\partial L}{\partial a_1} = \Sigma a_1 - 2\lambda a_1 = 0 \rightarrow (\Sigma - \lambda I) a_1 = 0 \quad (3.6)$$

El resultado es un sistema lineal de ecuaciones. Según el teorema de Rochè-Frobenius, para que el sistema tenga solución no nula o distinta de cero, la matriz $(\Sigma - \lambda I)$ tiene que ser singular. Esto implica que el determinante debe ser cero:

$$|\Sigma - \lambda I| = 0 \quad (3.7)$$

De modo que λ es un autovalor de Σ . La matriz de covarianzas Σ será de orden p y al definirla como positiva, p tendrá autovalores distintos, $\lambda_1, \lambda_2, \dots, \lambda_p$ tales que, por ejemplo, $\lambda_1 > \lambda_2 > \dots > \lambda_p$. Si se desarrolla la expresión anterior:

$$(\Sigma - \lambda I) a_1 = 0 \quad (3.8)$$

$$\Sigma a_1 - \lambda I a_1 = 0 \quad (3.9)$$

$$\Sigma a_1 = \lambda I a_1 \quad (3.10)$$

Entonces:

$$Var(y_1) = Var(a'_1 x) = a'_1 \sum a_1 = a'_1 \lambda a_1 = \lambda a'_1 a_1 = \lambda \cdot 1 = \lambda \quad (3.11)$$

Para maximizar la varianza de y_1 se toma el mayor autovalor λ_1 , y el autovector a_1 . Este a_1 se corresponde con un vector que devolverá la combinación de las variables originales con mayor varianza. De esta manera, si $a'_1 = (a_{11}, a_{12}, \dots, a_{1p})$, entonces:

$$y_1 = a'_1 x = a_{11}x_1 + a_{12}x_2 + \dots + a_{1p}x_p \quad (3.12)$$

El segundo de los componentes principales $y_2 = a'_2 x$, se obtiene de manera similar. Además, se requiere que y_2 esté incorrelado con el componente y_1 , es decir, $Cov(y_2, y_1) = 0$. Por lo tanto:

$$Cov(y_2, y_1) = Cov(a'_2 x, a'_1 x) = a'_2 \cdot E[(x - \mu)(x - \mu)] \cdot a_1 = a'_2 \sum a_1 \quad (3.13)$$

Esto es, se requiere que $a'_2 \Sigma a_1 = 0$. Como $\Sigma a_1 = \lambda a_1$, lo anterior equivale a:

$$a'_2 \Sigma a_1 = a'_2 \lambda a_1 = \lambda a'_2 a_1 = 0 \quad (3.14)$$

Y esto a su vez es equivalente a que $a'_2 a_1 = 0$, es decir, que los vectores sean ortogonales. De esta manera, hay que maximizar la varianza de y_2 , es decir, $a'_2 \Sigma a_2$, que está sujeta a las restricciones: $a'_2 a_2 = 1$ y $a'_2 a_1 = 0$.

Se toma entonces la función:

$$L(a_2) = a'_2 \sum a_2 - \lambda(a'_2 a_2 - 1) - \delta a'_2 a_1 \quad (3.15)$$

Se deriva y se multiplica por a'_1 :

$$\frac{\partial L(a_2)}{\partial a_2} = 2 \sum a_2 - 2\lambda a_2 - \delta a_1 = 0 \quad (3.16)$$

$$2a'_1 \sum a_2 - \delta = 0 \quad (3.17)$$

Despejando:

$$\delta = 2a'_1 \sum a_2 = 2a'_2 \sum a_1 = 0 \quad (3.18)$$

De este modo,

$$\frac{\partial L(a_2)}{\partial a_2} = 2 \sum a_2 - 2\lambda a_2 - \delta a_1 = 2 \sum a_2 - 2\lambda a_2 = \left(\sum -\lambda I \right) a_2 = 0 \quad (3.19)$$

Usando los mismos razonamientos que anteriormente, se elige λ como el segundo autovalor mayor de la matriz Σ y se asocia su autovector a a_2 .

De igual manera, al j-ésimo componente le correspondería el j-ésimo autovalor. Todos los componentes y se podrían expresar como el producto de una matriz formada por los autovectores, que es multiplicada por el vector x que a su vez contiene las variables originales, tal que $y=Ax$; donde:

$$y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_p \end{pmatrix}, \quad A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1p} \\ a_{21} & a_{22} & \cdots & a_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ a_{p1} & a_{p2} & \cdots & a_{pp} \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{pmatrix} \quad (3.20)$$

Como la varianza de cada y es igual a cada una de las respectivas λ , la matriz de covarianzas será:

$$\Delta = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_p \end{pmatrix} \quad (3.21)$$

porque y_1, \dots, y_p se han construido como variables incorreladas. Por tanto, se tiene que:

$$\Delta = Var(Y) = A'Var(X)A = A' \sum A \quad (3.22)$$

A modo de resumen, si se quiere calcular las componentes principales, se deben representar los datos como una matriz (siendo las filas las muestras y las columnas las variables), se resta la media para cada dimensión y se calculan los valores y vectores propios de la matriz de covarianza. Una vez hecho esto, se eligen los componentes y se forma una base con sus vectores, para luego ordenar los λ_i autovalores. Se comenzará eligiendo por el primero hasta conseguir que la suma de los restantes sea igual o menor que λ_i . Como la suma total de estos autovalores no es más que una expresión de la varianza de los datos, se puede utilizar un porcentaje de corte para desechar aquellas direcciones de dispersión que aportan menos cantidad de información.

3.3 Vectores comunes discriminantes

3.3.1 Introducción

Partiendo de la técnica del vector común, que extrae de un conjunto de datos aquél vector con respecto al cual varían los demás vectores de la muestra (es decir, el vector común), se puede plantear la técnica de los vectores comunes discriminantes. Esta última se cuestiona si puede existir una proyección a un subespacio en la cual tengan una única proyección todos los elementos de una misma clase que han sido proyectados en este subespacio. Esto quiere decir que en dicho subespacio estarán las muestras de una misma clase representadas por un único vector llamado vector común discriminante [20].

Lo que pretende la técnica de los vectores comunes es precisamente extraer las propiedades comunes de las clases en el entrenamiento, eliminando las diferencias de las muestras dentro de cada clase. De esta manera, se consigue obtener un vector común para cada una de las clases, de forma que se eliminan todas las componentes en la dirección de los autovectores correspondientes a autovalores no nulos de la matriz de dispersión total de su propia clase.

Para esta técnica, en lugar de usar la matriz de dispersión total intraclase, se utiliza la matriz de dispersión interclase. A partir de los vectores comunes se obtendrán un nuevo conjunto de vectores llamados vectores comunes discriminantes, que se usarán para la clasificación.

3.3.2 Cálculo de los DCV

Dado un conjunto de datos (en este caso un conjunto de entrenamiento) compuesto de C clases con N muestras cada clase, y siendo x_m^i un vector columna dimensional de dimensión d que denota a la muestra m -ésima de la clase i -ésima, habrá un total de $M = N \times C$ muestras. Asumiendo que $d > M - C$, las matrices de dispersión intraclase S_w e interclase S_b , vienen dadas por:

$$S_w = \sum_{i=1}^C \sum_{m=1}^N (x_m^i - \mu_i)(x_m^i - \mu_i)^T \quad (3.23)$$

$$S_b = \sum_{i=1}^C N(\mu_i - \mu)(\mu_i - \mu)^T \quad (3.24)$$

Y la matriz total, que es suma de las otras dos, será:

$$S_T = \sum_{i=1}^C \sum_{m=1}^N (x_m^i - \mu)(x_m^i - \mu)^T \quad (3.25)$$

Donde μ se corresponde con la media de las muestras, y μ_i la media de las muestras de la clase i -ésima y se define como:

$$\mu_i = \frac{1}{N} \sum_{m=1}^N x_m^i \quad y \quad \mu_i = \frac{1}{M} \sum_{i=1}^C \sum_{m=1}^N x_m^i \quad (3.26)$$

Si se busca hallar los vectores de la proyección óptima w en el espacio nulo de S_w , se necesita encontrar el conjunto de vectores ortonormales que abarca el espacio nulo de S_w [21]. Sin embargo, es complicado abordar esta tarea intratable desde el punto de vista computacional dado que puede ser demasiado grande la dimensión de dicho espacio nulo. Dicha tarea se puede llevar a cabo de manera más eficiente utilizando el complemento ortogonal de S_w , que se corresponde con un espacio vectorial de dimensión menor.

Se supone R^d el espacio de muestra original, V como el espacio de rango de S_w y V^\perp el espacio nulo de S_w , esto es:

$$V = span\{\alpha_k | S_w \alpha_k \neq 0, k = 1, \dots, r\} \quad (3.27)$$

$$V^\perp = span\{\alpha_k | S_w \alpha_k = 0, k = 1, \dots, r\} \quad (3.28)$$

Donde $r < d$ se corresponde con el rango de S_w , $\{\alpha_1 \dots \alpha_r\}$ con un conjunto ortonormal, y $\{\alpha_{r+1} \dots \alpha_d\}$ con el conjunto de autovectores ortonormales que corresponden a los autovalores no nulos de S_w .

Se consideran entonces las matrices $Q = \alpha_1 \dots \alpha_r$ y $\bar{Q} = \{\alpha_{r+1} \dots \alpha_d\}$. Como $R^d = V \oplus V^\perp$, cualquier vector $x_m^i \in R^d$ se descompone de la siguiente manera:

$$x_m^i = y_m^i + z_m^i \quad (3.29)$$

Donde:

$$y_m^i = P x_m^i = Q Q^T x_m^i \in V \quad (3.30)$$

$$z_m^i = \bar{P} x_m^i = \bar{Q} \bar{Q}^T x_m^i \in V^\perp \quad (3.31)$$

El vector único para las muestras de la misma clase vendría dado por:

$$x_{com}^i = x_m^i - Q\bar{Q}^T x_m^i = \bar{Q}\bar{Q}^T x_m^i \quad (3.32)$$

De esta manera, se obtiene el vector común para todas las muestras de la misma clase. Los vectores de proyección óptima serían aquellos que maximizan la dispersión total de los vectores comunes. En otras palabras:

$$\begin{aligned} J(W_{opt}) &= \arg \max |w^T S_w w| = \arg \max |w^T S_B w| \\ &= \arg \max |w^T S_{com} w| \quad \text{si } |w^T S_w w| = 0 \end{aligned} \quad (3.33)$$

Donde W es una matriz en la que las columnas se corresponden con los vectores de proyección óptimos ortonormales w_k , y S_{com} es:

$$S_{com} = \sum_{i=1}^C (x_{com}^i - \mu_{com})(x_{com}^i - \mu_{com})^T, \quad i = 1, \dots, C \quad (3.34)$$

Siendo μ_{com} la media de los vectores comunes. Los vectores de proyección óptima w_k pueden ser hallados mediante un análisis propio de S_{com} . Todos los vectores propios que se corresponden con los valores propios diferentes de cero de S_{com} serán vectores de proyección óptima. S_{com} se puede definir como una matriz grande de dimensión $d \times d$, y a partir de esto se podrá usar la matriz menor, $A_{com}^t A_{com}$ de tamaño $C \times C$, para hallar los valores propios y los vectores propios de $S_{com} = A_{com}^t A_{com}$, donde A_{com} se corresponde con una matriz de $d \times C$ de la forma:

$$A_{com} = x_{com}^1 - \mu_{com} \dots x_{com}^C - \mu_{com} \quad (3.35)$$

Dado que los vectores de proyección óptima w_k están en el espacio nulo de S_w , en el momento en que los vectores de muestra x_m^i de la clase i -ésima sean proyectados en el espacio creado por los vectores w_k , el vector de características de los coeficientes de proyección no dependerá del índice de la muestra m . Por lo que para cada clase:

$$\Omega_i = W^T x_m^i \quad (3.36)$$

A estos vectores se les denomina vectores comunes discriminantes. El hecho de que los coeficientes de proyección no dependan del índice m garantiza que la precisión en el reconocimiento de muestras esté en torno al 100%, para el conjunto de entrenamiento.

El vector de características utilizado para reconocer una imagen de test será:

$$\Omega_{test} = W^T x_{test} \quad (3.37)$$

En resumen, se puede estructurar el método DCV en los siguientes pasos [10]:

1. Calcular los valores propios no nulos y los correspondientes vectores propios w_k de S_w usando la matriz $A^t A$. Establecer $Q = [\alpha_1 \dots \alpha_r]$, donde r se corresponde con la dimensión de S_w .
2. Escoger una muestra de cada clase y proyectarlas todas sobre el espacio nulo de S_w de manera que se puedan obtener los vectores comunes, según la ecuación (3.32).
3. Calcular los componentes principales mediante los vectores propios w_k de S_{com} , que corresponden a los valores propios distintos de cero usando la matriz $A_{com}^t A_{com}$.
4. Elegir un vector de características que representa cada una de las clases y proyectarlo con la matriz W . El resultado obtenido será un vector de clasificación, y la proximidad con este vector será la que determine la pertenencia o no a la clase de la muestra analizada.

3.4 Transformada discreta del coseno

3.4.1 Introducción

La transformada discreta del coseno es una herramienta que puede ser utilizada en procesamiento digital para reconocer patrones y para la compresión de imágenes [22].

La DCT [23] es una transformada similar a la transformada discreta de Fourier, con una diferencia fundamental: la DCT elimina los coeficientes senoidales de la transformación. De esta manera, se tiene una transformada real, dado que los vectores base se componen tan solo de funciones coseno muestreadas. La DCT tiene una buena capacidad de compactación de la energía al dominio transformado, es decir, que logra concentrar la mayor parte de la información en pocos coeficientes transformados.

Esto último resulta útil para el reconocimiento de patrones, ya que se puede calcular la DCT para retener un subconjunto de los coeficientes que conforman el vector de características usado para describir el espacio de imagen [24].

3.4.1 Cálculo de la DCT

Teniendo una secuencia de entrada $u(n)$ de longitud N , su DCT vendrá dada por [25]:

$$v(k) = \alpha(k) \sum_{n=0}^{N-1} u(n) \cos\left(\frac{(2n+1)\pi k}{2N}\right) \quad 0 \leq k \leq N-1 \quad (3.38)$$

Donde:

$$\alpha(0) = \sqrt{\frac{1}{N}}, \quad \alpha(k) = \sqrt{\frac{2}{N}} \quad 1 \leq k \leq N-1 \quad (3.39)$$

La secuencia $u(n)$ se puede interpretar como un vector y la DCT como una matriz de transformación que se aplica a este vector. Esta matriz de transformación se expresa como:

$$c(k, n) = \begin{cases} \frac{1}{\sqrt{N}} & k = 0, \quad 0 \leq n \leq N-1 \\ \sqrt{\frac{2}{N}} \cos\left(\frac{(2n+1)\pi k}{2N}\right) & 1 \leq k \leq N-1, \quad 0 \leq n \leq N-1 \end{cases} \quad (3.40)$$

Donde k y n son respectivamente los índices de las filas y las columnas. También se puede obtener la inversa de la DCT a partir de su salida, de modo que:

$$u(n) = \sum_{k=0}^{N-1} \alpha(k) \cos\left(\frac{(2n+1)\pi k}{2N}\right) \quad 0 \leq n \leq N-1 \quad (3.41)$$

En definitiva, al aplicar la DCT a una secuencia de entrada, ésta se puede descomponer en una suma pesada de secuencias básicas de coseno en la que los componentes con mayor energía se ubican en las primeras posiciones. De esta manera, se usarán estos coeficientes para construir el vector de características que permita diferenciar una imagen de otra.

3.5 Resumen

La extracción de características es fundamental para transformar un conjunto de características originales, como pueden ser los píxeles de cada imagen, en un conjunto de características de menor tamaño. La extracción de características

permite reducir la dimensionalidad de los datos, mejora la tasa de generalización y disminuye el coste computacional del sistema de cara a la clasificación. Para realizar esta tarea se han utilizado tres técnicas diferentes.

La primera, la de Análisis de Componentes Principales (PCA), se basa en realizar una reducción de dimensión manteniendo la información intrínseca de los datos de entrada para reducir el coste computacional del sistema en la etapa de clasificación.

El método de Vectores Comunes Discriminantes (DCV) extrae las características comunes de cada una de las clases analizadas mediante la eliminación de las diferencias entre las muestras de cada clase de manera que el vector de características contenga las características comunes de las clases.

La tercera técnica usada, la de la Transformada Discreta del Coseno (DCT). Esta genera un vector de características formado por los coeficientes de la DCT y cada elemento es una representación de una frecuencia de la imagen. Los componentes que tengan mayor energía se colocan en las primeras posiciones y serán los que se usen para construir el vector de características.

Capítulo 4: Clasificación

4.1 Introducción

El método de clasificación utilizado en este proyecto será el de las Máquinas de Vectores Soporte (SVM). Esta técnica tiene su origen en los estudios sobre la teoría del aprendizaje estadístico introducidos en los años 90 por Vapnik y sus colaboradores [26].

Originariamente las SVMs fueron pensadas para resolver problemas de clasificación binaria. Sin embargo, se utilizan en la actualidad para resolver otros tipos de problemas (regresión, agrupamiento, multclasificación). Han sido utilizadas con éxito también en otros campos, como visión artificial, reconocimiento de caracteres, categorización de texto e hipertexto o análisis de series temporales.

En este capítulo se plantea esta técnica de clasificación para realizar un reconocimiento automático [27] a partir de los vectores de características.

4.2 Máquinas de vectores soporte

Las SVM son modelos de aprendizaje supervisados con algoritmos de aprendizaje asociados que analizan los datos utilizados para la clasificación y el análisis de regresión.

Dado un conjunto de ejemplos de entrenamiento, cada uno marcado como pertenecientes a una u otra de las dos categorías, un algoritmo de entrenamiento de SVM [28] construye un modelo que asigna nuevos ejemplos a una categoría u otra, convirtiéndolo en un clasificador binario no probabilístico lineal. De otro modo, dado un conjunto separable de ejemplos $S = \{ (x_1, y_1) \dots (x_n, y_n) \}$, donde $x_i \in \mathbb{R}^d$, e $y_i \in \{+1, -1\}$, se puede definir un hiperplano de separación como una función lineal que puede separar dicho conjunto sin error:

$$D(\mathbf{x}) = (w_1x_1 + \dots + w_dx_d) + b = \langle \mathbf{w}, \mathbf{x} \rangle + b \quad (4.1)$$

Donde w y b son coeficientes reales. El hiperplano de separación cumpliría las siguientes restricciones:

$$\begin{aligned} \langle w, x_i \rangle + b &\geq 0 & \text{si } y_i = +1 \\ \langle w, x_i \rangle + b &\leq 0 & \text{si } y_i = -1, i = 1, \dots, n \end{aligned} \quad (4.2)$$

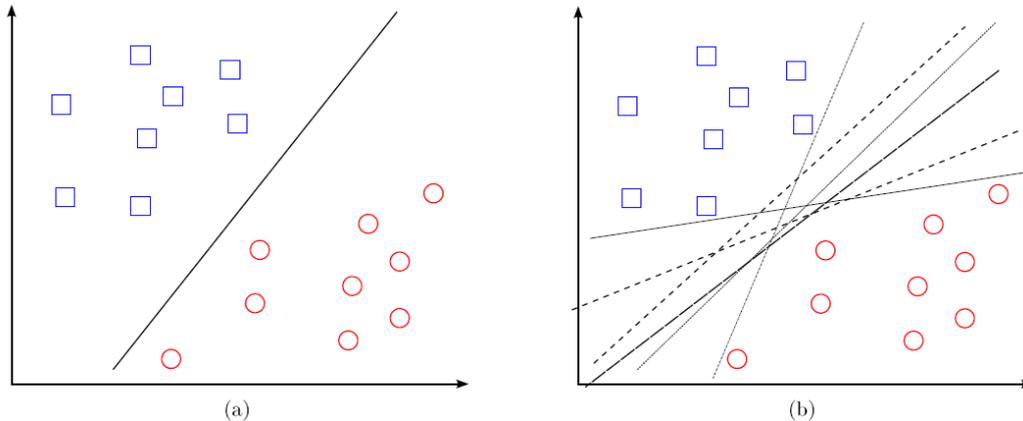


Fig. 15: Hiperplanos de separación en un espacio bidimensional:
 (a) ejemplo de hiperplano de separación
 (b) otros ejemplos de hiperplanos de separación, de entre los infinitos posibles.

La Fig.15 es un claro ejemplo de clasificador lineal, puesto que un clasificador de este tipo separa un conjunto de objetos pertenecientes a dos clases en sus respectivos grupos mediante una línea. Sin embargo, no todas las tareas de clasificación son tan simples, y se hace necesario en la mayoría de ellas el uso de estructuras más complejas, o el uso de más de un plano de separación. En el caso de este proyecto, la tarea de clasificación según las muestras de entrenamiento, requerirá de estructuras más avanzadas de separación, basada en múltiples hiperplanos de separación.

Una propiedad de la definición de hiperplano de separación óptimo es que éste mantiene equidistancia respecto al ejemplo más cercano de cada clase. Esta propiedad se puede demostrar de manera sencilla mediante reducción al absurdo. Suponiendo que la distancia del hiperplano óptimo al ejemplo más cercano de la clase “+1” fuese menor que la del ejemplo más cercano de la clase “-1”, se podría alejar el hiperplano del ejemplo de la clase “+1” una distancia mayor que la anterior y que al mismo tiempo siga siendo menor que la distancia al ejemplo más cercano de la clase “-1”. De esta manera, se puede aumentar el tamaño del margen cuando al inicio se supuso que éste era el hiperplano óptimo o máximo. Se puede aplicar un razonamiento análogo al suponer que la distancia del hiperplano óptimo al ejemplo más cercano de la clase “-1” fuese menor que la del ejemplo más cercano de la clase “+1”.

Más formalmente, un SVM construye un hiperplano o conjunto de hiperplanos en un espacio de alta o infinita dimensión, que puede usarse para la clasificación, regresión u otras tareas. Intuitivamente, se logra una buena separación por el hiperplano que tiene la mayor distancia al punto de datos de entrenamiento más

próximo de cualquier clase (llamado margen funcional), ya que en general, cuanto mayor es el margen, menor es el error de generalización del clasificador.

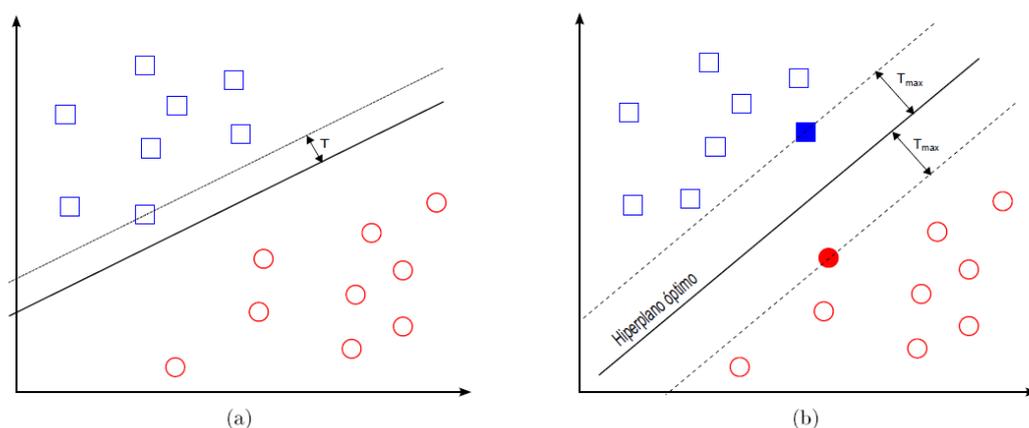


Fig. 16: Margen de un hiperplano de separación:
(a) hiperplano de separación no-óptimo, (b) hiperplano de separación óptimo

En resumen, las Máquinas de Vectores soporte (SVM) cumplen las tareas de clasificación a partir de la construcción de hiperplanos en un espacio de varias dimensiones, que separan muestras etiquetadas de distintas clases.

4.2.1 Modos de funcionamiento de SVM

Se pueden distinguir dos modos básicos de funcionamiento de las Máquinas de Vectores Soporte.

Por un lado, el modo de entrenamiento, en el que debe ser entrenado el sistema con el objetivo de crear un modelo característico que será usado en la fase de clasificación o test. Con el fin de crear dicho modelo, el sistema recibe los vectores de características como patrones de entrada para configurar una salida con resultado conocido. Este resultado es introducido por etiquetas, que indican la clase a la que cada muestra pertenece. Esta técnica se conoce como entrenamiento supervisado y por cada SVM utilizado se genera un modelo diferente.

En cuanto al modo de test, el sistema funciona de manera diferente. Ya generados los modelos en la fase de entrenamiento, una vez se introduce una muestra desconocida, el sistema debe ser capaz de clasificar cada muestra de acuerdo a la clase que pertenece de las diferentes presentes en el conjunto de la base de datos.

4.3 Uso de SVM en tareas de identificación

La identificación o reconocimiento consiste en averiguar si algo está dentro de un grupo de cosas que ya son conocidas. En el presente caso, interesaría saber a qué especie de araña de las contenidas en la base de datos pertenece cada muestra del conjunto de test.

Como se ha visto en la sección anterior, las Máquinas de Vectores Soporte están fundamentadas en un sistema de clasificación de dos clases. En este proyecto se está trabajando con más clases diferentes, como suele suceder en la mayoría de los casos que involucran reconocimiento de patrones. Por tanto, para solucionar este problema, hay que utilizar no solamente una, sino varias Máquinas de Vectores Soporte, como mínimo tantas como clases diferentes existan.

Para seleccionar el número de SVM que se va a emplear hay que elegir en primer lugar la estrategia de clasificación que se va a emplear. Existen numerosas estrategias, pero dentro de ellas se van a estudiar dos de cara a la viabilidad de su implementación. Estas son [29]: la estrategia uno contra uno (“one versus one”) y la estrategia uno contra al resto (“one versus all”).

En la estrategia uno contra uno, se utiliza una SVM para cada pareja de clases que se pueda configurar. De esta manera, para cada SVM se genera un modelo que permite comparar dos clases concretas, resaltando las diferencias entre ambas. Así, por ejemplo, para un problema de cuatro clases (1, 2, 3 y 4) se emplearán las SVM con entradas: 1-2, 1-3, 1-4, 2-3, 2-4 y 3-4. De esta manera se obtienen seis modelos diferentes donde cada uno corresponde a cada una de las SVM y con esto se consigue que todas las clases sean comparadas entre sí.

Por otro lado, en la estrategia uno frente al resto, cada clase se enfrenta a todas demás, que serán agrupadas en una sola clase como si fueran la misma. Esto es, cada clasificador SVM determinara si una muestra pertenece a una clase determinada o si por el contrario pertenece a alguna de las tres restantes. En este caso, el número de modelos y de SVM utilizados coincide con el número de clases. Considerando el ejemplo expuesto anteriormente, se utilizarán cuatro Máquinas de Vectores Soporte distribuidas de la siguiente manera: la 1 contra el resto, la 2 contra el resto, la 3 contra el resto y la 4 contra al resto.

Entre ambas estrategias, para este proyecto se seleccionará la última, debido a que optimiza el número de modelos generados y además la capacidad de discriminación es mayor, lo que nos permite obtener mejores resultados.

4.4 Uso de SVM en tareas de verificación

La verificación es el proceso mediante el que se comprueba si algo o alguien es quien dice ser. Aplicado en este proyecto, dado que se tienen cuatro clases diferentes, verificar una de las muestras del conjunto de test consiste en

comprobar que corresponde a la especie de araña a la que pertenece dicha muestra y no a cualquiera de las otras.

Dos conceptos básicos en las tareas de verificación son la tasa de verdaderos positivos y la tasa de falsos positivos. La tasa de verdaderos positivos se basa en la probabilidad que tiene el sistema de autenticar adecuadamente la especie a la cual pertenece la muestra de la telaraña que se está analizando en el conjunto de test. Por el contrario, la tasa de falsos positivos se basa en la probabilidad del sistema de identificar de manera errónea la especie a la cual pertenece dicha muestra.

Partiendo de estos parámetros probabilísticos, se construyen las curvas ROC (Característica Operativa del Receptor). Dichas curvas se construyen a partir de la representación de la razón de verdaderos positivos (VPR) en el eje y frente a la razón de falsos positivos (FPR) en el eje x, según se varía el umbral de discriminación [30].

Una vez construidas estas curvas, se puede fijar el método de predicción óptimo en un punto en la esquina superior izquierda [10] del espacio ROC; es decir, aquel punto en el que no hay ningún falso negativo ni ningún falso positivo. Esto daría como resultado una clasificación perfecta, con el 100% de verdaderos positivos y el 0% de falsos positivos. En el polo opuesto, una clasificación errónea o aleatoria se situaría en un punto a lo largo de una línea diagonal construida desde el extremo inferior izquierdo hasta la esquina superior derecha. Esta línea también se suele llamar que línea de no-discriminación,

A partir de la curva ROC se pueden generar diferentes estadísticos que son susceptibles de ser usados para analizar el rendimiento del clasificador. Uno de los más utilizados y que se va a emplear en este proyecto es el del área bajo la curva ROC, también denominado como AUC. Este estadístico se puede definir como la probabilidad de que un clasificador ordene una instancia positiva elegida al azar con una puntuación más alta que una negativa. De esta manera se recurre a una medida global que es independiente del punto de corte con las líneas de discriminación.

Para seleccionar entre dos pruebas distintas se compara entre el AUC obtenido en cada una de las pruebas. Comúnmente el AUC tiene un valor que se encuentra comprendido entre 0,5 y 1, donde el 1 arroja como resultado una verificación del 100% o perfecta y donde el 0,5 representa a un test sin capacidad discriminatoria o dicho con otras palabras, un test poco acertado o aleatorio. Es decir, si el AUC para una prueba es 0.9 significa que existe un 90% de probabilidad de que el sistema obtenga una clasificación correcta de una de las muestras, escogidas al azar entre las del vector de muestras de test. Por esto, el sistema que presenta una mayor área bajo la curva será el que se identifique con el más efectivo o con mejor capacidad discriminadora [31]. La Fig.17 detalla las diferentes curvas de ROC según el valor del AUC.

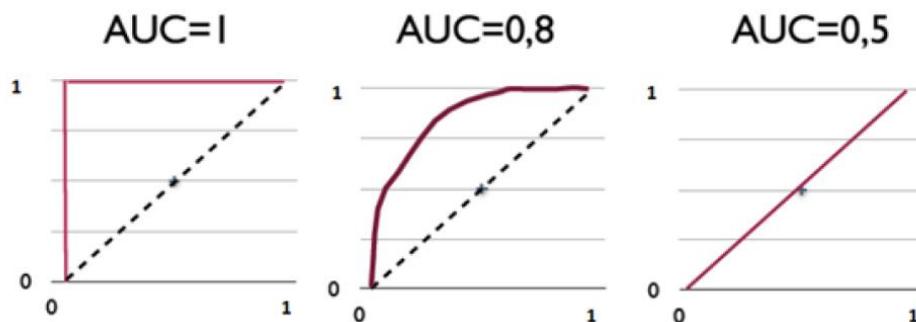


Fig. 17: Curvas ROC según el valor AUC

Para analizar los resultados experimentales se tendrá en cuenta que, si el valor del AUC está entre 0.97 y 1, el test se calificará como excelente. Entre 0.9 y 0.97 el test se calificará de muy bueno, entre 0.75 y 0.9 se calificará como bueno, entre 0.6 y 0.75 como regular y finalmente si el valor del AUC está entre 0.5 y 0.6, el test será malo.

En cuanto al funcionamiento del clasificador en tareas de verificación, la fase de entrenamiento y las características del sistema son iguales que para las tareas de identificación, a excepción de la estructura para modo test que se explicará a continuación.

El procedimiento que describe el funcionamiento del clasificador en modo test consiste en introducir un vector de características perteneciente a una muestra del conjunto de test en una de las SVM entrenada previamente y en la que los vectores de características de la clase a la que pertenece la muestra que se va a evaluar han sido etiquetados con "+1". Si la salida supera un cierto umbral, se puede concluir que el vector de características en la entrada corresponde a la clase que se pretende verificar.

Al igual que para la identificación, en la verificación se utilizará el clasificador de cuatro clases basado en la estrategia de uno frente al resto. De esta manera, también se utilizará en verificación cuatro SVM y cuatro modelos diferentes obtenidos durante la fase de entrenamiento.

4.5 Módulos de fusión

Los resultados individuales obtenidos en identificación y verificación para cada uno de los sistemas pueden ser mejorados si se combinan las salidas obtenidas para cada uno de estos sistemas individuales. Es aquí donde entra la posibilidad de utilizar diferentes métodos de fusión para conseguir esta combinación, de modo que el sistema final sea más robusto y el porcentaje de acierto en identificación y verificación sea mayor que los obtenidos con cada uno de los sistemas por separado.

Dada las características de este proyecto, y comprobada su funcionalidad en trabajos anteriores [10], se utilizarán los módulos de decisión del “más votado” y el módulo “adding scores” [32].

El método de fusión del más votado se basa en obtener la salida de cada uno de los sistemas y a partir de estas, elegir aquella en la que más sistemas coinciden. La “salida más votada” será aquella por tanto que tiene más probabilidades de ser la correcta, a partir de las evaluadas por los sistemas individuales. Para implementar esta técnica, se almacena en un vector de salidas el resultado de cada uno de los sistemas individuales y se calcula la moda de ese vector para determinar cuál es la clase más votada.

Por otro lado, el método “adding scores” consiste en realizar la suma de los valores obtenidos en cada una de las salidas para cada SVM y después seleccionar como válida la clase que obtenga la puntuación más alta.

En las siguientes figuras se presentan los diagramas de bloques de ambos módulos:

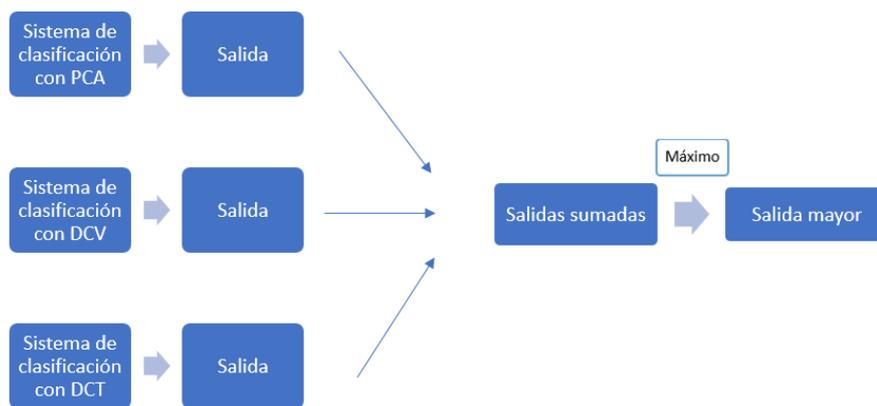


Fig. 18: Módulo de fusión "Adding scores"

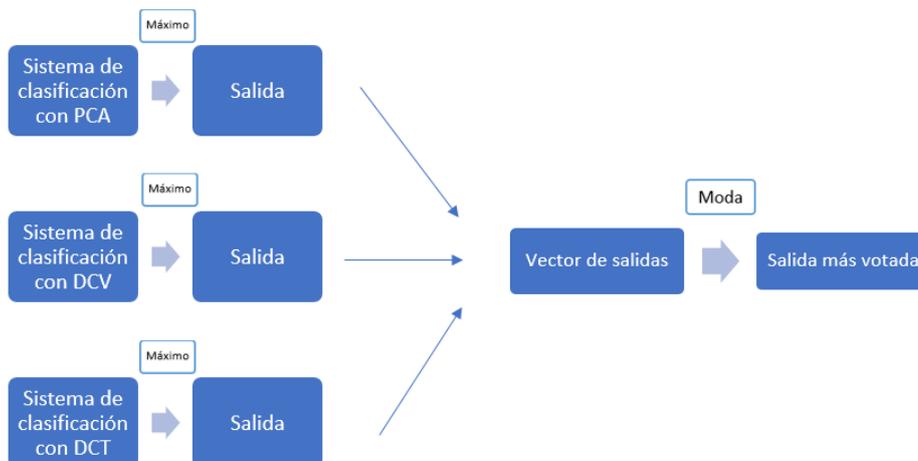


Fig. 19: Módulo de fusión "más votado"

4.6 Resumen

El modo de clasificación que se ha implementado en el actual proyecto se basa en las Máquinas de Vectores Soporte o SVM. Con esta técnica se pretende realizar a partir de los vectores de características una clasificación basada en aprendizaje supervisado con algoritmos que analizan los datos utilizados para clasificar diferentes clases, en este caso aquellas que son objeto de estudio del proyecto, las muestras de telaraña de diferentes especies.

El sistema de clasificación usual basado en SVM es en un sistema que solo discrimina entre dos clases a la entrada del clasificador. Dado que hay más de una clase que clasificar, deben usarse varias SVM, en concreto cuatro, ya que se emplea la estrategia de uno frente al resto, en la que cada clase es comparada con el resto que es agrupada en una sola clase contrapuesta a la primera.

Se distinguen dos modos de funcionamiento del clasificador, el modo de entrenamiento y el modo de test. Se llevarán a cabo a partir de estos modos tanto tareas de reconocimiento o identificación como tareas de verificación.

Las tareas de reconocimiento o identificación consisten en averiguar si la muestra de telaraña que se presenta en el modo de test corresponde a alguna de las especies contenidas en la base de datos. Por otro lado, la verificación se basa en comprobar que dicha muestra de test corresponde verdaderamente a la especie de araña original a la que dice pertenecer y no a cualquiera de las otras.

Por último, se utilizarán los módulos de fusión de sistemas para implementar un sistema más robusto y con mejores resultados. En concreto se implementarán el método del Más Votado y el método "Adding scores".

Capítulo 5: Metodología experimental

5.1 Introducción

Una vez han sido estudiados los fundamentos teóricos en los que se basa el presente proyecto, en este capítulo se presentará la metodología que se ha seguido a lo largo de la fase experimental. Se va a exponer de manera detallada cada uno de los bloques o pasos que se han seguido, desde la obtención de la base de datos ya preprocesada hasta los sistemas de clasificación usados.

Es importante tener en cuenta que el sistema que se utilizará para la clasificación requiere que el conjunto total de datos sea dividido en dos subconjuntos, uno para entrenar el sistema y otro de test para evaluar su eficacia y rendimiento. En la próxima sección se detallará los procesos a seguir para la división de este conjunto de datos.

5.2 Estructura del sistema

El desarrollo experimental de este proyecto se apoya sobre un sistema que cuenta con varios módulos bien diferenciados.

En primer lugar, se encuentra el módulo de preprocesado, que se encarga de realizar las transformaciones necesarias a las imágenes originales de la base de datos para que estas presenten un óptimo estado de contraste y aislamiento de la tela de araña de cara a la extracción de características.

En segundo lugar, la base de datos preprocesada se expondrá a un sistema de validación de resultados que pasa por crear el modelo que distribuirá aleatoriamente las muestras en dos subconjuntos, el de entrenamiento y el de test.

En tercer lugar, se extraerán las características de los vectores que han sido obtenidos en las dos etapas anteriores y que contienen las particularidades de

los conjuntos de imágenes de la base de datos. Como ya se ha explicado, se utilizarán las técnicas de análisis de componentes principales (PCA), vectores comunes discriminantes (DCV) y transformada discreta del coseno (DCT).

Finalmente se halla el módulo que contiene el clasificador, que como su propio nombre indica, se encarga de clasificar los vectores obtenidos en la fase de extracción de características. El método utilizado para la clasificación está basado en Máquinas de Vectores Soporte.

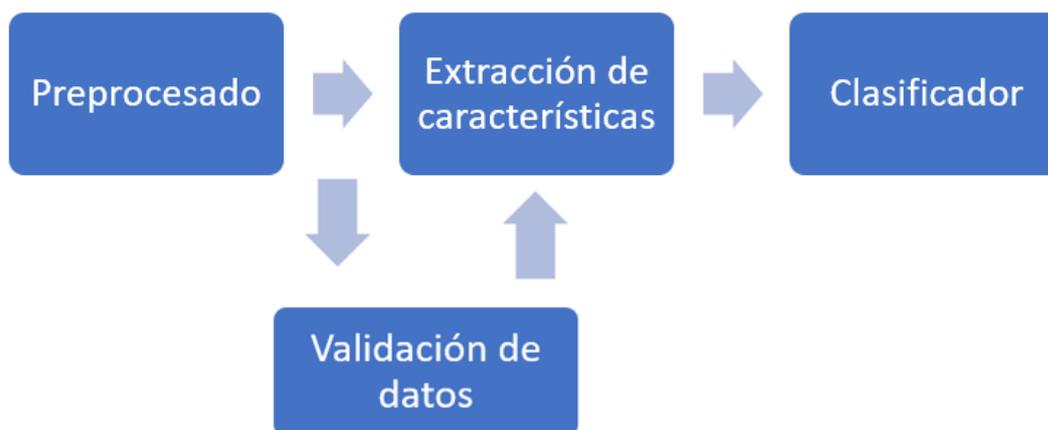


Fig. 20: Diagrama de bloques del sistema experimental

5.2.1 Etapa de preprocesado

La base de datos debe estar preparada para realizar una extracción de características correcta. Para hacer esto, se trata la base de datos original con los métodos explicados en el capítulo 2.

Como resultado, se tienen dos bases de datos de imágenes segmentadas, una usando el método Otsu y otra usando el método Canny. Como ya se ha comentado, estudiando la colección de las imágenes, se puede observar que en el centro de la telaraña es donde se presentan las mayores diferencias. Por lo tanto, se puede determinar que dicho centro contiene una parte importante diferenciadora de la información. De modo que es interesante construir una nueva base de datos para cada método, cortando las imágenes para aislar el centro. Finalmente, se consigue una base de datos con imágenes de la telaraña completa y del centro para el método Otsu y otra análoga para el método de Canny.

En el método de Otsu, se observa que algunas de las imágenes (un 40% aproximadamente) no presentan resultados óptimos al aplicarse el umbral automáticamente calculado por la función "graythresh" debido a la excesiva iluminación en las imágenes originales, puesto que identificaba demasiados píxeles como blancos. Un ejemplo de esto se presenta en las siguientes

imágenes preprocesadas mediante este método (para la telaraña completa y para el centro de la tela), perteneciente a la clase Eustala illicita.

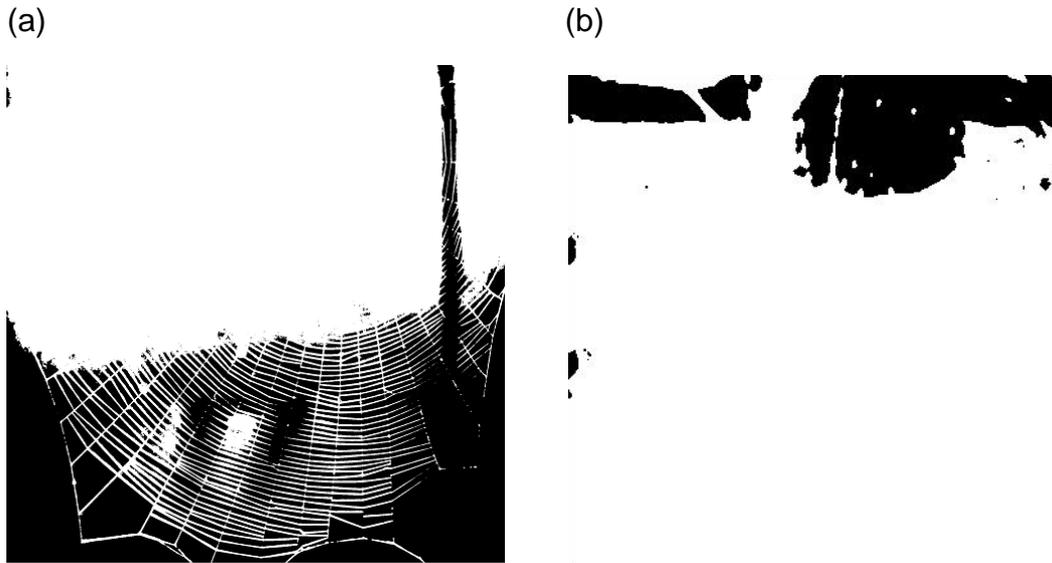


Fig. 21: Imágenes umbralizadas de forma automática
(a) Telaraña completa, (b) centro de la telaraña

Para paliar este problema y poder disponer de una base de datos más adecuada para la extracción de características se realizó un ajuste semi-automático del umbral, de manera que los píxeles seleccionados como blancos en la binarización no superaran un 40% del total de píxeles de la imagen, mediante la función *selector.m*. Esta función identifica el número de píxeles en blanco y si estos superan el 40% del total, el umbral se aumenta en pasos centesimales hasta que la imagen binarizada cumpla ese requisito del 40%. Este reajuste tiene la desventaja de que oscurece ciertas partes de la telaraña, pero no se llega a perder la silueta de la misma, tal y como se aprecia en las siguientes imágenes:

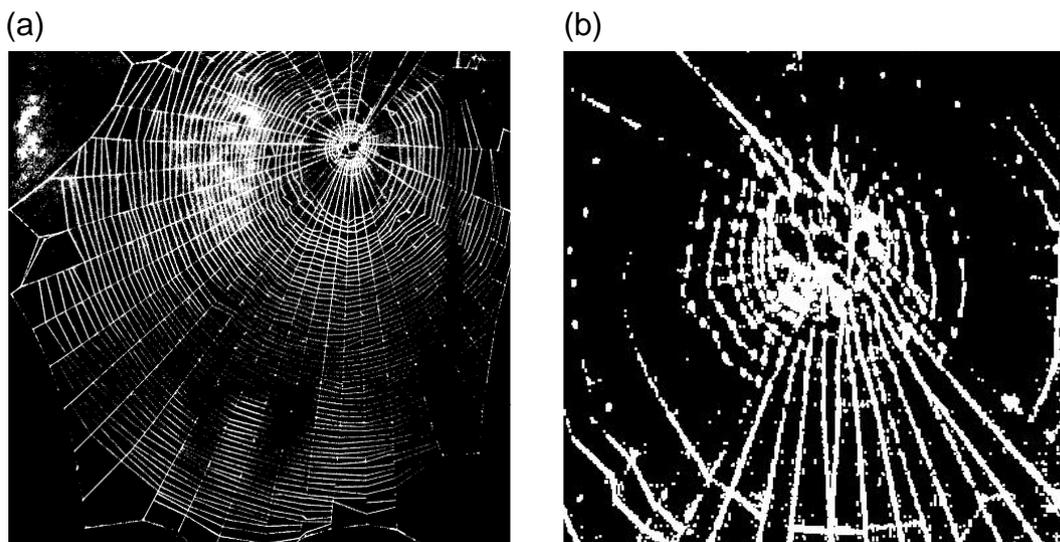


Fig. 22: Imágenes procesadas con ajuste de umbralización
(a) Telaraña completa, (b) centro de la telaraña

En cuanto al método de Canny, tomando la misma imagen que en el caso anterior, el proceso que se ha seguido arroja los siguientes resultados:

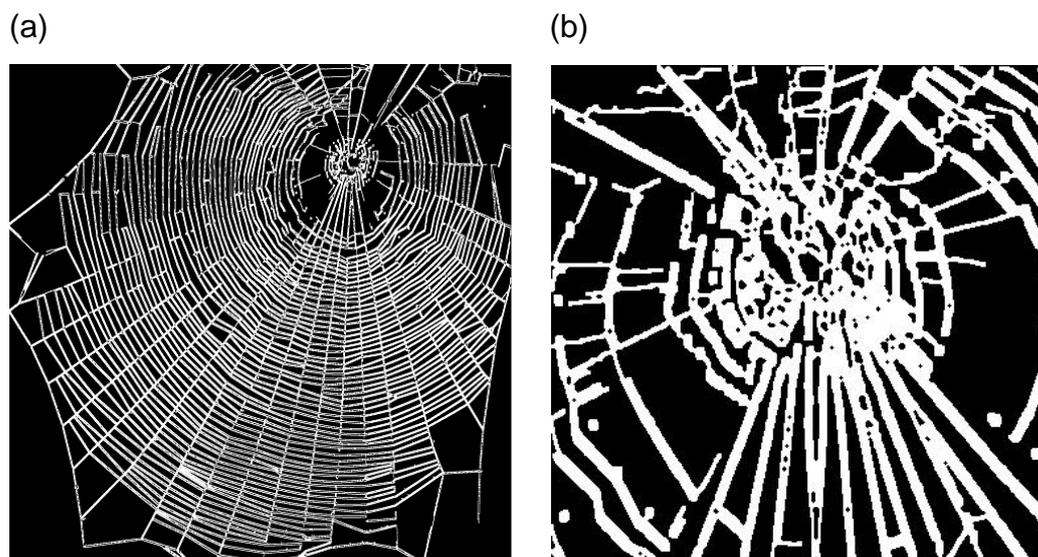


Fig. 23: Imágenes preprocesadas por el método de Canny
(a) Telaraña completa, (b) centro de la telaraña

Visualmente, el resultado final parece ser más adecuado en el método de Canny. No obstante, se utilizan las dos bases de datos de ambos métodos en la clasificación y se comparan los resultados obtenidos.

Ahora bien, para poder analizar las imágenes de forma automática, estas deben redimensionarse a un determinado ancho y alto. Una vez hecho esto, se concatenan las filas y columnas de las imágenes redimensionadas, para formar un vector de grandes dimensiones. El número de píxeles total del vector resultante se conoce también como espacio de imagen.

Para que los experimentos fueran independientes de la adquisición de las imágenes, y con el fin de encontrar la longitud óptima para conseguir el redimensionamiento de cara a una mejor extracción de características y clasificación, las imágenes se redimensionan a diferentes tamaños. Se comenzará por un tamaño de 5x5 píxeles, hasta llegar a un tamaño de 50x50 píxeles, en saltos de 5 en 5.

Tamaño de la imagen (píxeles)	5x5	10x10	15x15	20x20	25x25	30x30	35x35	40x40	45x45	50x50
Espacio de imagen (píxeles)	25	100	225	400	625	900	1225	1600	2025	2500

Tabla 2: Tamaños de redimensionado de imágenes y de espacio de imagen

La base de datos reducida se obtiene a partir de un script de Matlab en el que las imágenes son leídas con la función “*imread*” y se vectorizan gracias a “*imreshape*”, incluidas en la *Image Processing Toolbox* de Matlab. El resultado se almacena en un archivo “.*mat*”, de manera que para cada tamaño de redimensionado se tendrá una base de datos simplificada y vectorizada. Cada archivo contiene una matriz en la cual las filas serán las imágenes y las columnas el espacio de las mismas [10].

5.2.2 Validación de datos

Una vez se ha estructurado la base de datos reducida tanto para las muestras con segmentación basada en el método de Otsu como para las muestras con segmentación basada en el método de detector de bordes de Canny, el siguiente paso consiste en extraer las características de dicha base de datos. Para todo el proceso restante se pretende medir el rendimiento alcanzado por el algoritmo de aprendizaje que se va a utilizar. Para ello, cada conjunto de datos se divide a su vez en dos subconjuntos: uno de entrenamiento y otro de prueba, como ya se ha explicado en apartados anteriores.

Para la división en estos dos subconjuntos, la regla que suele aplicarse en los procesos de identificación y reconocimiento es usar el 50% de la base de datos para el aprendizaje de patrones o entrenamiento y el 50% restante para realizar las pruebas o test. Dicha división es aconsejable que sea realizada al azar, de manera que se asegure que los resultados sean más genéricos y no dependan de una colocación determinada de las muestras de la base de datos, sino que la distribución sea aleatoria. Al tener cuatro clases, lo óptimo es que las muestras de cada clase se dividan al 50% en los conjuntos de entrenamiento y test, de manera que haya muestras de todas las clases en ambos conjuntos en la misma proporción.

Para la base de datos actual, compuesta de 185 muestras, una posible distribución podría ser la que se muestra en la siguiente tabla:

Conjunto	Eustala ilícita (76 imágenes)	Cyclosa caroli (38 imágenes)	Nephila edulis (33 imágenes)	Araneus diadematus (38 imágenes)
Entrenamiento	38	19	17	19
Test	38	19	16	19

Tabla 3: Distribución de las muestras en conjuntos de entrenamiento y test

Los algoritmos de entrenamiento buscan los patrones utilizando el conjunto de entrenamiento y puede darse que algunos de estos patrones descubiertos sean falsos o bien solo sean válidos para un conjunto determinado de entrenamiento y no para el sistema general. Esto se debe principalmente a que las muestras seleccionadas para el entrenamiento han sido elegidas al azar. Por ello, si se pretende realizar medidas precisas, se deben realizar no sobre este conjunto de entrenamiento, sino sobre un conjunto de prueba que sea independiente. Para asegurar por tanto que los resultados sean óptimos, se hace uso de diferentes técnicas de validación cruzada [34].

En este proyecto se utilizan dos métodos de validación cruzada, que se pasarán a explicar a continuación. Estos son el método de validación Hold-Out y la validación cruzada de k-iteraciones o k-fold.

El método de *Hold-Out*, también conocido como validación cruzada aleatoria, es el método más simple de los dos presentados y también el más rápido computacionalmente hablando. Este método consiste en dividir al azar el conjunto de datos de entrenamiento y el conjunto de datos de prueba, de manera que las muestras no se solapen [36], es decir, que no se repitan en ambos conjuntos para la misma iteración. Para cada división se puede ajustar el porcentaje para el cual se quiere disponer en número de muestras para entrenamiento y para test. En el caso que nos ocupa, se utilizará Hold-Out del 50%, del 60%, del 70%, del 80% y del 90% en el conjunto de test, y se analizarán las prestaciones dependiendo de dicho porcentaje. Un ejemplo gráfico de cómo funcionaría este método de validación se presenta en la siguiente figura:

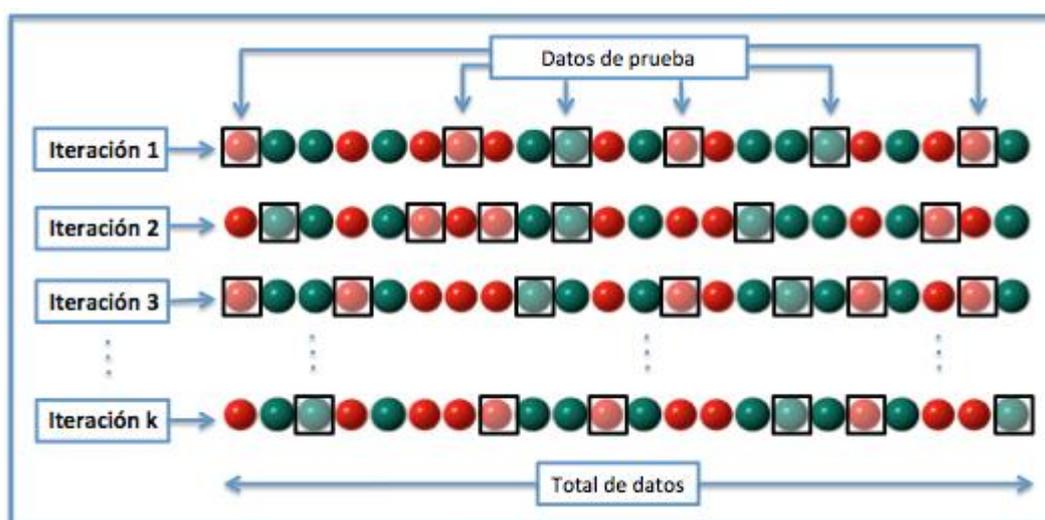


Fig. 24: Validación Hold-Out

Una sola validación cruzada de las muestras no es concluyente, ya que se pueden obtener diferentes resultados dependiendo de cómo se distribuyen las muestras para pruebas y entrenamiento. Es decir, entre una y otra simulación pueden darse diferencias sustanciales debido a la aleatoriedad de la distribución de muestras. Es por ello que, para realizar un sistema más robusto y fiable, se realizarán 10 iteraciones usando el método Hold-out para cada simulación y a la hora de presentar los experimentos se realizará la media de los resultados obtenidos en las 10 iteraciones.

El método de validación cruzada de k-iteraciones [36] cumple también el objetivo que se persigue para la validación de datos. El procedimiento usado en esta técnica consiste en dividir los datos de muestra en k subconjuntos, de los cuales solo uno de ellos se usa para test. El resto, es decir, $k-1$ subconjuntos serán utilizados como muestras para entrenamiento. Con la intención de que cada uno de los subconjuntos puedan ser tratados como muestras para test, el proceso de

validación cruzada se irá repitiendo un total de k iteraciones. Finalmente se calcula la media aritmética de los resultados obtenidos en cada iteración para obtener un único resultado. Este método tiene la ventaja frente a Hold-Out de que es más preciso, porque asegura que todas las muestras pasen por el subconjunto de prueba en la misma simulación. Sin embargo, presenta una desventaja, ya que es mucho más lento computacionalmente que Hold-Out, debido precisamente a que se deben realizar k -iteraciones por simulación. Un ejemplo gráfico de este método se presenta en la figura 25, para un valor de $k=5$.

De igual manera que con el método Hold-Out, con el método de validación cruzada de k -iteraciones se repetirá el proceso un total de 10 veces para asegurar la fiabilidad del sistema.

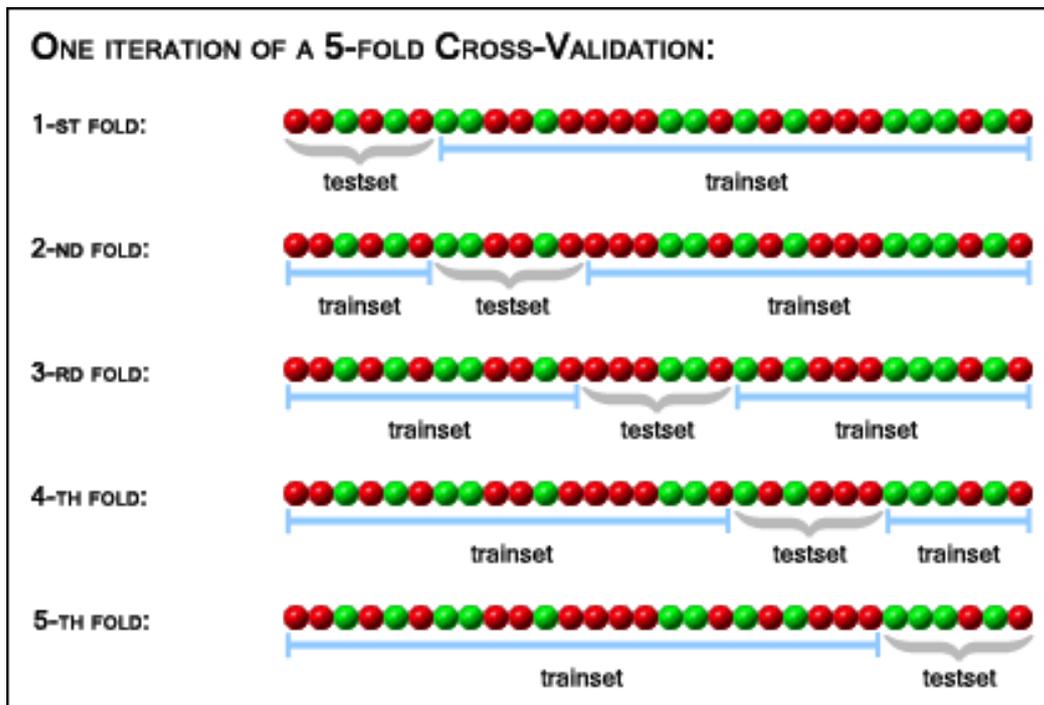


Fig. 25: Validación k -iteraciones

5.2.3 Extracción de características

Una vez que las muestras de la base de datos se han distribuido correctamente entre el entrenamiento y las muestras de prueba, éstas se encuentran adecuadas y normalizadas para poder ser tratadas a lo largo del resto de módulos que componen el sistema.

El siguiente módulo experimental es el de extracción de características, el cual se encarga de reducir el amplio número de características de las imágenes a un número menor de componentes o coeficientes de manera que la clasificación no sea tan pesada computacionalmente. Aunque la dimensionalidad de los datos debe reducirse, esto se tiene que realizar de manera que las características sean discriminantes entre una y otra clase para que la clasificación sea óptima. El

proceso de extracción de características se implemente utilizando diferentes técnicas cuyo planteamiento teórico ha sido explicado en el capítulo 3. Estas son: PCA, DCV y DCT.

5.2.3.1 Aplicación de PCA

Como ya se ha visto, una de las características principales del análisis de componentes principales (PCA) es la capacidad de disminuir la dimensión de los datos para conseguir un vector de características más reducido que el original, lo que hace de PCA una herramienta importante para tratar problemas de dimensionalidad [18]. Para conseguir este vector de características reducidas, el proceso a seguir consiste en extraer del conjunto de vectores de una misma clase un número determinado de componentes, llamados principales.

En este proyecto, este número está determinado por el tamaño del espacio de las imágenes que como ya se ha visto depende del tamaño de redimensionamiento que se utilice. Por ejemplo, para una imagen que ha sido redimensionada a un tamaño de 10x10 píxeles, su espacio de imagen tendrá un tamaño de 100 píxeles, por lo que el vector final de componentes principales tendrá 100 componentes.

Para aplicar la técnica de análisis de componentes principales, los datos se distribuyen de acuerdo a lo expuesto en la sección 5.2.1.1 en el que se explica la estructura de la base de datos reducida de manera que el número de filas es equivalente al número de muestras de entrenamiento de las que se dispone, mientras que el espacio de imagen corresponderá con las columnas.

Una vez se aplica PCA se dispondrá por un lado de la matriz de entrenamiento transformada, y, por el otro, la matriz de proyección que se usa para proyectar las muestras durante el entrenamiento. Con esta última matriz se proyectan a continuación los datos del conjunto de test, de manera que tanto este como el de entrenamiento queden dispuestos en el mismo espacio. Ambos conjuntos serán los usados para la entrada de datos de las diferentes SVM; el de entrenamiento se usará para la generación de patrones de reconocimiento y el de test para probar el rendimiento del sistema.

Una vez que se han realizado las transformaciones sobre los conjuntos de datos, se realizará un número determinado de simulaciones, ejecutando un barrido para buscar este número óptimo de componentes, mediante el método del análisis a saltos que será realizado sobre la cantidad de componentes principales. Finalmente, el número óptimo de coeficientes será aquel para el que mejores resultados se hayan obtenido durante el proceso de clasificación.

5.2.3.2 Aplicación de DCV

La técnica DCV (explicada en el *Capítulo 3.3*) consiste en representar cada clase con un vector común discriminante mediante la proyección de las muestras en un espacio superior. DCV es una técnica supervisada, lo que implica que debe

indicarse a que clase pertenece cada una de las muestras para un correcto funcionamiento del algoritmo.

En la técnica de vectores comunes determinantes, el número de componentes vendrá fijado por el algoritmo que calcula los vectores comunes, de manera heurística. Comúnmente, el número de componentes suele coincidir con el número de clases existentes o una unidad menos.

5.2.3.3 Aplicación de DCT

En el método DCT, a través de la transformada de coseno discreta, cada transformación se realiza para cada muestra o imagen independientemente del resto. Mediante la aplicación de esta transformación, se consigue una reducción de la información utilizando los primeros coeficientes obtenidos, que representan los valores de energía más elevados [29].

De forma similar a PCA, se aplica la DCT unidimensional [28], a cada fila de la matriz de manera que el número de filas se corresponde con el número de muestras de entrenamiento y las columnas con el espacio de imagen. Por lo que se realizarán en este ejemplo tantas transformadas DCT unidimensionales independientes como número de muestras del conjunto de entrenamiento

El número de coeficientes óptimos de la Transformada Discreta del Coseno se calcula de manera similar a PCA, esto es, analizar la cantidad de coeficientes utilizando el método conocido como a saltos.

5.2.4 Clasificación utilizando SVMs

Una vez que las muestras de entrenamiento y test contenidas en los vectores de entrada son analizadas en la fase de extracción de características, los vectores de características resultantes serán introducidos en el clasificador construido con Máquinas de Vector Soporte.

Como ya se ha explicado en el capítulo 4.3, la estrategia elegida para la clasificación de las muestras es la llamada "*uno contra todos*". Esta estrategia compara cada clase con todas las demás contenidas dentro de la misma clase, por lo que existen tantos modelos como clases. En el caso que nos ocupa, se dispone de cuatro SVMs, que calcularan el valor para cada clase enfrentada al resto.

Se va a distinguir qué sistema de clasificación se utilizó diferenciando si la tarea de clasificación estaba destinada a identificación o si estaba destinada a verificación.

5.2.4.1 Tareas de identificación

En tareas de identificación, los vectores de características se introducen en las cuatro SVMs implementadas, asignando etiquetas positivas o negativas para identificar qué clase se está evaluando frente al resto. A modo de ejemplo, en la primera SVM, se asigna la etiqueta "+1" a los vectores de características que corresponden a la primera clase, y se asigna la etiqueta "-1" a los vectores de características del resto de clase. Y con las demás SVM se procede de manera análoga, de manera que lo único que cambia es la distribución de las etiquetas, permitiendo que se trabaje siempre con los mismos vectores de características. Un esquema de este proceso se muestra en la siguiente figura:

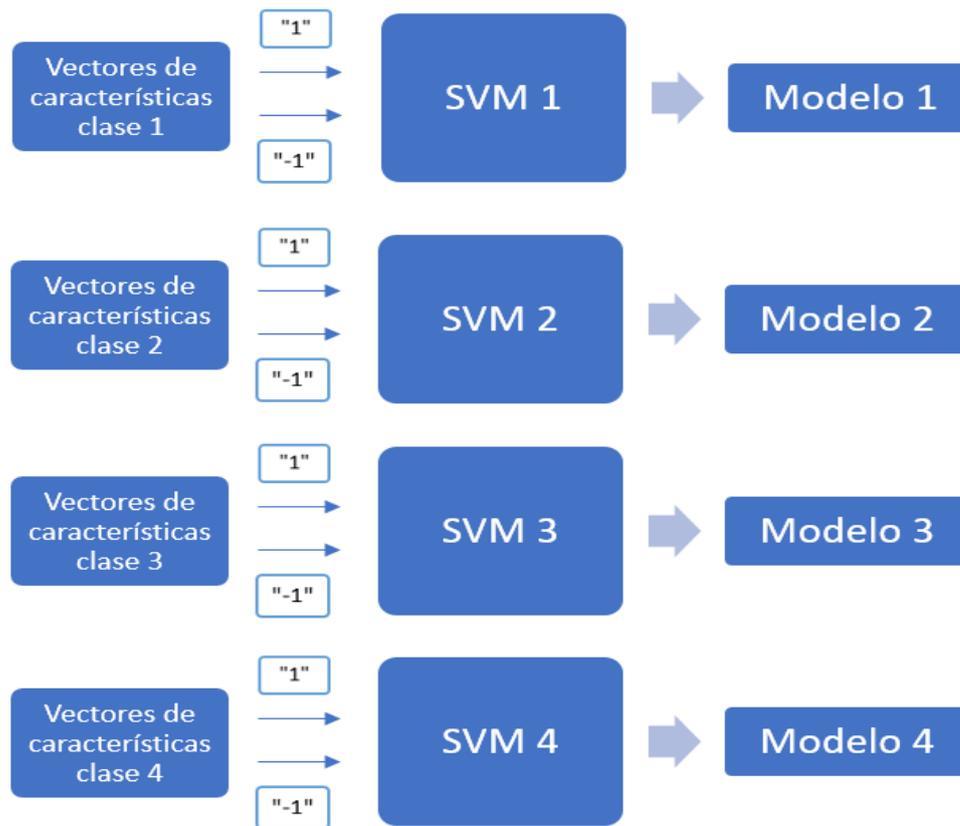


Fig. 26: Diagrama de bloques del clasificador SVM para entrenamiento con 4 clases

A partir de cada SVM se genera un modelo diferente (cuatro en total). Los modelos se utilizarán para analizar los vectores de características del conjunto de muestras de test, de manera que se compruebe la capacidad de discriminación del clasificador. Una vez se ha introducido el vector de características de test en cada uno de los modelos, el resultado se almacena en un vector de salidas, donde cada salida corresponde con un valor indicativo de la similitud del vector de características estudiado con respecto a cada una de las clases. El valor máximo obtenido en este vector de salidas indicará con qué clase tiene más similitud el vector de características estudiado. Este proceso se resume en la figura 27.

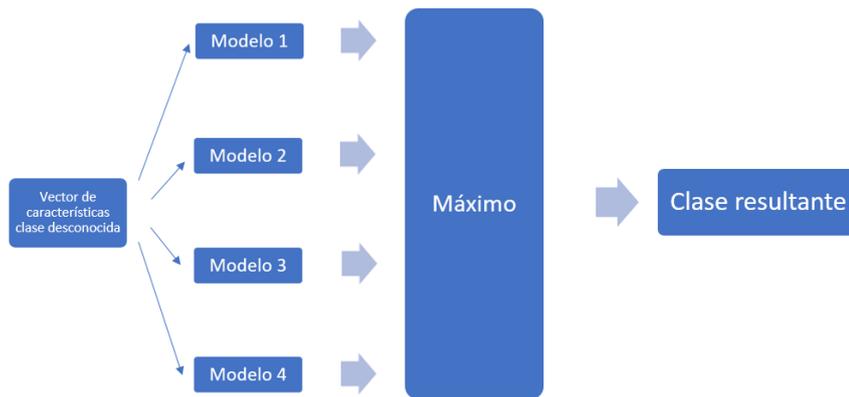


Fig. 27: Diagrama de bloques del clasificador SVM para test con 4 clases

Por último, utilizando una matriz de confusión, se comparan los resultados obtenidos en cada muestra con la clase a la que realmente pertenece, para calcular la tasa de éxito del clasificador.

5.2.4.2 Tareas de verificación

La metodología que se seguirá para la verificación presentará algunas diferencias en el proceso respecto a la metodología usada en las tareas de identificación.

En primer lugar, y esto sí es análogo a las tareas de identificación, se entrenan las diferentes SVM del sistema. En segundo lugar, la imagen que se quiere clasificar se somete al preprocesado explicado el capítulo 3 este proyecto. En tercer lugar, se extraen las características de la muestra, calculando de esta manera un vector de características, que se introducirá en el modelo obtenido del entrenamiento de la SVM que se corresponde con la clase que quiere verificarse. En el caso de que la salida supere un determinado umbral, la muestra será verificada como positiva o perteneciente a la clase a la que dice pertenecer. En caso contrario, la verificación será negativa. La siguiente figura muestra el diagrama de bloques de este proceso de verificación:



Fig. 28: Diagrama de bloques del clasificador para la verificación de la clase

Al igual que en tareas de identificación, se utiliza una matriz de confusión, pero en esta ocasión fijando el umbral y calculando a partir de los resultados obtenidos la tasa de verdaderos positivos y de falsos positivos.

5.2.4.3 Parámetros a tener en cuenta para la clasificación

Un factor importante que se debe tener en cuenta radica en que para la implementación del clasificador se ha usado el paquete de funciones LS-SVM de MATLAB, de manera que se trabaja con una etapa de clasificación funcional y previamente utilizada con resultados óptimos [10].

Basándonos en la estructura de este paquete se puede establecer dos parámetros fundamentales que deben ser configurados para una buena construcción del clasificador. Estos parámetros son el parámetro de coste c y la clase de función kernel que se utiliza.

Empezando por el último, se utilizará una función kernel de base radial gaussiana (RBF), ya que experimentalmente ofreció mejores resultados que la de base lineal. Esta función viene definida por [28]:

$$K(x, y) = e^{-\frac{\|x-y\|^2}{2\sigma^2}} \quad (5.1)$$

Donde σ^2 es la varianza del kernel gaussiano. Esta varianza se debe ajustarse de manera que su valor sea inversamente proporcional a la media de las muestras. Este parámetro es el más importante a la hora de realizar el ajuste, ya que es el que presenta un mayor efecto en la tasa de acierto en identificación.

El segundo parámetro que debe ajustarse es el parámetro c , que se corresponde con el coste del entrenamiento. No es tan sencillo determinar el valor adecuado para el parámetro c debido a que este viene dado comúnmente por los datos de entrada del clasificador y su ajuste se debe implementar de forma automática.

Experimentalmente, tanto el parámetro σ^2 como el parámetro c han sido calculados de manera automática por el sistema en cada experimento.

5.3 Resumen

La estructura global del sistema utilizado en el proceso experimental se divide en varios módulos cuyos fundamentos teóricos han sido explicados en los anteriores capítulos.

En primer lugar, se encuentra un módulo de preprocesado, que consiste en preparar las imágenes de modo que solo contengan la zona de interés de la telaraña, de manera que puedan ser correctamente analizadas por módulos posteriores.

A continuación, y con el objetivo de comprobar la eficiencia y fiabilidad de los sistemas de extracción de características y clasificación, se realizará una validación cruzada de las muestras de la base de datos reducida, mediante las

técnicas de Hold-Out y k-iteraciones, que consiste en dividir el conjunto de datos en un subconjunto de muestras para entrenamiento y otro subconjunto de muestras para test.

Una vez divididas las muestras, se llega al módulo de extracción de características, basado en tres técnicas diferentes: PCA, DCV y DCT. Este módulo analiza las muestras de los conjuntos de entrenamiento y test de la base de datos para construir otros vectores de menor tamaño que permitan diferenciar una muestra de otra con mayor precisión.

El último módulo usado es el clasificador, basado en SVM, que se basa en clasificar los vectores de características obtenidos para analizar posteriormente cada muestra del conjunto de test y ser capaz de identificarla o verificarla dentro de una u otra clase. Estas clases han sido analizadas y diferenciadas gracias a la fase de entrenamiento del clasificador.

Capítulo 6: Resultados experimentales

6.1 Introducción

En el presente capítulo se van a exponer los diferentes experimentos realizados y los resultados obtenidos en los mismos, de manera que se pueda comparar qué técnicas de preprocesado facilitan un mejor funcionamiento y precisión de los sistemas de identificación y verificación, así como aquellos métodos de extracción de características que mejores resultados arrojan.

Para cada experimento se han llevado a cabo diferentes simulaciones, teniendo en cuenta los dos pares de bases de datos que resultaron del preprocesado para los métodos de segmentación de Otsu y Canny y del recorte de la base de datos con la telaraña completa para obtener las muestras del centro de la telaraña.

Una vez obtenidos los resultados, se podrán extraer las conclusiones y comprobar si se cumplen los objetivos del proyecto, tal y como se plantearon en el Capítulo 1.

6.2 Simulaciones realizadas

Uno de los primeros experimentos realizados ha consistido en determinar el tamaño óptimo de redimensionamiento de imagen, es decir, aquel tamaño con el que mejores resultados se consiguen. El interés de estas pruebas radica en la capacidad del sistema de encontrar el tamaño adecuado, debido a que si se redimensiona a un tamaño pequeño se puede reducir la información de la imagen. Por el contrario, un tamaño de imagen demasiado grande podría contener mucha información y dificultar la extracción de características, lo que podría reducir de forma considerable la tasa de acierto del clasificador.

De forma análoga, se realizaron las simulaciones para hallar el número de coeficientes de los sistemas basados en PCA y DCT, así como los resultados obtenidos automáticamente para cada tamaño de imagen en DCV.

Para realizar estas simulaciones se han realizado probaturas en primer lugar el método Hold-Out con un porcentaje del 50% para el conjunto de entrenamiento, con 10 iteraciones, luego con 20 y finalmente con 30. Los resultados no varían significativamente aumentando el número de iteraciones a partir de 10, así que se estima que 10 iteraciones son suficientes para comprobar la fiabilidad del sistema y computacionalmente el coste es menor.

Finalmente, se realizarán 10 iteraciones usando la validación cruzada de k -iteraciones para distintos valores típicos de k (3, 5, 7 y 10), con el fin de analizar la fiabilidad y robustez del sistema cuando se utiliza este tipo de validación.

6.3 Resultados obtenidos en identificación

En esta sección se procederá a detallar los experimentos realizados y los resultados obtenidos en identificación.

En primer lugar, y como se ha explicado en el apartado anterior, se obtendrá el tamaño óptimo de redimensionamiento de las imágenes mediante un análisis a saltos sobre el número de píxeles (ancho por alto). Se presentarán los resultados para la base de datos compuesta por imágenes del centro de la telaraña, así como para la compuesta de imágenes de la telaraña completa. De igual manera, también se presentarán los resultados para cada uno de los métodos de segmentación utilizados en el preprocesado. Como ya se ha explicado, se utilizará para estas simulaciones el método de validación cruzada Hold-Out al 50% para el conjunto de entrenamiento

Posteriormente, se mostrarán los resultados de los sistemas basados en PCA, DCT y DCV, con el número de coeficientes o componentes óptimos para cada uno. Por último, se mostrarán los resultados de la fusión de sistemas mediante los métodos del Más Votado y "Adding scores".

6.3.1 Tamaño y número de coeficientes para la telaraña completa

Con el objetivo de encontrar el tamaño óptimo al de redimensionamiento de las imágenes y el número de coeficientes óptimo, se realizó una serie de simulaciones mediante la técnica de análisis a saltos. Como se explicó anteriormente, las simulaciones se realizarán ejecutando 10 iteraciones con el método de validación cruzada Hold-Out con un porcentaje del 50% sobre el conjunto de entrenamiento.

En la tabla 4 se resumen los resultados obtenidos a partir de todas las simulaciones realizadas. El método seguido para la representación de los resultados consiste en mostrar para cada base de datos (la basada en el método de Otsu y la basada en el método de Canny) y para cada método de extracción de características, la tasa de acierto del sistema. La tasa de acierto mostrada para cada sistema es aquella que corresponde a las simulaciones realizadas con el número óptimo de coeficientes o componentes principales. No está de más

recordar que los tamaños de redimensionado se han obtenido desde un tamaño inicial de 5x5 píxeles hasta un tamaño final de 50x50 píxeles, en saltos de cinco en cinco.

(a)

Sistema	Tasa de acierto máxima por cada tamaño de redimensionamiento				
	5x5	10x10	15x15	20x20	25x25
PCA (Otsu)	77,35%	78,90%	78,72%	78,28%	77,38%
PCA (Canny)	81,51%	82,72%	81,65%	81,06%	81,23%
DCV (Otsu)	71,26%	71,58%	71,77%	71,75%	71,62%
DCV (Canny)	72,77%	72,95%	73,33%	73,28%	73,49%
DCT (Otsu)	74,21%	75,56%	75,65%	75,44%	77,62%
DCT (Canny)	77,94%	78,81%	79,37%	79,87%	80,92%

(b)

Sistema	Tasa de acierto máxima por cada tamaño de redimensionamiento				
	30x30	35x135	40x40	45x45	50x50
PCA (Otsu)	79,45%	78,59%	77,36%	76,45%	74,01%
PCA (Canny)	83,21%	81,36%	80,35%	79,27%	76,80%
DCV (Otsu)	71,65%	71,53%	71,38%	71,11%	70,91%
DCV (Canny)	73,01%	72,75%	72,40%	72,07%	71,66%
DCT (Otsu)	76,84%	76,28%	75,72%	74,69%	72,05%
DCT (Canny)	81,24%	79,95%	78,15%	76,91%	73,84%

Tabla 4: Resultados para el tamaño óptimo de redimensionado de la telaraña completa

Para determinar el tamaño de redimensionado óptimo se observa la media de los resultados obtenidos de cada sistema por cada tamaño. Se ve por tanto que el tamaño óptimo de redimensionado es de 30x30 píxeles.

En la tabla 5 se presenta para cada método qué número de coeficientes ha sido el óptimo. En la sección correspondiente al análisis de los resultados obtenidos en cada método individual se detallará la variación de la tasa de acierto en relación al número de coeficientes usado.

Sistema	Nº de coeficientes óptimos para cada tamaño de redimensionamiento									
	5x5	10x10	15x15	20x20	25x25	30x30	35x35	40x40	45x45	50x50
PCA (Otsu)	10	10	10	10	10	10	10	10	10	15
PCA (Canny)	10	10	10	10	10	10	10	10	15	15
DCV (Otsu)	4	4	4	3	4	4	4	3	3	3
DCV (Canny)	4	3	3	3	4	4	4	3	3	3
DCT (Otsu)	10	50	40	50	70	70	90	100	100	100
DCT (Canny)	10	50	50	50	70	80	90	100	100	100

Tabla 5: Resultados para los coeficientes de los métodos de extracción de características de la telaraña completa

6.3.2 Tamaño y número de coeficientes para el centro de la tela

La obtención del tamaño de dimensionado óptimo para la base de datos del centro de la telaraña ha sido determinado de manera análoga a cómo se

procedió en el caso de la telaraña completa. Se han realizado las simulaciones correspondientes con el método de Hold-Out al 50% de entrenamiento.

En la siguiente tabla se muestran los resultados obtenidos para la base de datos que contiene las imágenes del centro de la telaraña.

(a)

Sistema	Tasa de acierto máxima por cada tamaño de redimensionamiento				
	5x5	10x10	15x15	20x20	25x25
PCA (Otsu)	81,96%	82,81%	83,31%	83,39%	84,11%
PCA (Canny)	85,55%	87,29%	86,52%	84,65%	87,66%
DCV (Otsu)	74,23%	74,57%	74,77%	74,79%	73,24%
DCV (Canny)	74,30%	75,01%	75,26%	75,66%	75,71%
DCT (Otsu)	76,61%	77,53%	78,55%	79,60%	80,43%
DCT (Canny)	83,21%	84,11%	85,79%	82,53%	86,48%

(b)

Sistema	Tasa de acierto máxima por cada tamaño de redimensionamiento				
	30x30	35x35	40x40	45x45	50x50
PCA (Otsu)	84,23%	83,62%	80,46%	80,25%	77,52%
PCA (Canny)	89,24%	84,91%	81,44%	81,32%	78,14%
DCV (Otsu)	73,79%	72,95%	72,83%	71,61%	70,90%
DCV (Canny)	75,25%	74,69%	73,35%	72,78%	72,45%
DCT (Otsu)	81,13%	81,06%	80,45%	79,22%	74,83%
DCT (Canny)	87,68%	85,31%	80,92%	79,77%	75,03%

Tabla 6: Resultados para el tamaño óptimo de redimensionado del centro de la telaraña

Como se puede observar si se comparan estas últimas tablas con las presentadas en el caso de la base de datos con la telaraña completa, los resultados en identificación mejoran bastante en el caso del estudio del centro de la telaraña, lo que soporta la hipótesis establecida sobre que es en el centro de la tela donde se encuentra la mayor parte de la información diferenciadora entre clases. También cabe destacar que el tamaño óptimo de redimensionamiento vuelve a ser de 30x30 para la base de datos con las muestras del centro de la telaraña.

Sistema	Nº de coeficientes óptimos para cada tamaño de redimensionamiento									
	5x5	10x10	15x15	20x20	25x25	30x30	35x35	40x40	45x45	50x50
PCA (Otsu)	5	10	10	10	10	10	10	10	10	10
PCA (Canny)	5	10	10	10	10	10	10	10	10	10
DCV (Otsu)	3	4	4	3	4	4	4	4	3	4
DCV (Canny)	3	3	4	3	3	4	4	3	3	4
DCT (Otsu)	10	40	80	50	100	90	90	120	130	130
DCT (Canny)	15	40	80	50	100	100	80	120	120	120

Tabla 7: Resultados para los coeficientes de los métodos de extracción de características del centro de la telaraña

En la tabla anterior se ha mostrado el número de coeficientes óptimo para cada método de extracción de características para la base de datos formada por las muestras del centro de la telaraña.

6.3.3 Resultados obtenidos por los sistemas individuales

Una vez se ha determinado cuál es el tamaño óptimo de redimensionado y el número de coeficientes para cada método que ofrece los mejores resultados, se tendrán en cuenta esos parámetros para analizar el rendimiento de los sistemas individuales basados en PCA, DCV y DCT.

De igual manera, como ya se ha demostrado que la base de datos que ofrece las tasas de acierto más altas es aquella que contiene las imágenes del centro de la telaraña, el análisis del rendimiento de cada sistema se realizará teniendo en cuenta solo esta base de datos, para optimizar el número de simulaciones.

Para cada sistema se realizarán primero simulaciones basadas en 10 iteraciones usando el método de validación cruzada Hold-Out, variando el porcentaje del conjunto de entrenamiento desde el 50% hasta el 10% en pasos de diez, para comprobar en qué grado disminuye la tasa de acierto a medida que se usan menos muestras para entrenar.

Para comprobar la robustez del sistema, también se llevarán a cabo simulaciones basadas en 10 iteraciones usando el método de validación cruzada k-iteraciones, con valores de k: 3, 5, 7 y 10.

6.3.3.1 Sistema basado en PCA

Antes de mostrar los resultados obtenidos realizando simulaciones con validación cruzada Hold-Out y k-iteraciones como se ha detallado en la anterior sección, resulta interesante mostrar cómo varía la tasa de aciertos del sistema basado en PCA respecto al número de componentes principales utilizados. Se recuerda que el tamaño de redimensionamiento óptimo era de 30x30 píxeles, lo que se traduce en un espacio de imagen de 900 píxeles.

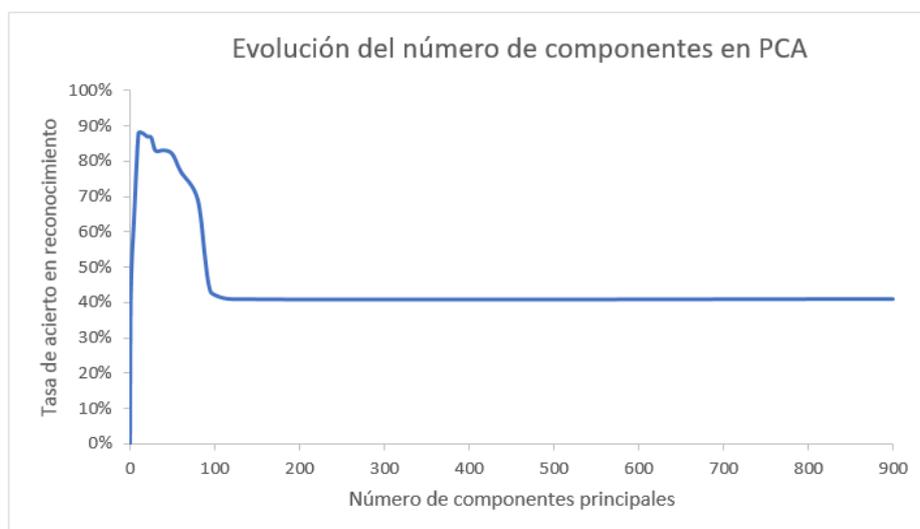


Fig. 29: Evolución del número de componentes principales del sistema PCA

La fig.29 se ha obtenido a partir de las simulaciones realizadas para el método de Canny. Al interpretarla, se comprueba cómo la tasa máxima de acierto se consigue con 10 componentes principales y a partir de ahí, la tasa de acierto va descendiendo de manera progresiva hasta estabilizarse en un 41% aproximadamente, que es la tasa mínima de acierto del sistema basado en PCA.

Según lo mostrado en la tabla 8, se consigue el mejor resultado con un 90,31% utilizando la validación 10k-iteraciones, ya que con esta técnica se dispone de un conjunto mayor de muestras para el entrenamiento. Al dividir las muestras en diez conjuntos, se usarán en cada iteración diez para entrenamiento y uno para test.

Los resultados de las simulaciones se presentan en la siguiente tabla:

Hold-Out (% de test)	Tasa de acierto del sistema basado en PCA	
	Método de Otsu	Método de Canny
50%	84,23%	89,24%
60%	83,15%	85,48%
70%	81,52%	82,91%
80%	74,61%	75,98%
90%	65,30%	67,57%
k-iteraciones	Método de Otsu	Método de Canny
10	87,64%	90,31%
7	85,93%	89,57%
5	85,54%	88,96%
3	84,12%	88,25%

Tabla 8: Resultados obtenidos con PCA

En cuanto a la técnica de validación Hold-Out, se observa que si bien el porcentaje de acierto desciende a medida que se usan menos muestras para entrenar (y por tanto más muestras para test), solo baja del 80% cuando el porcentaje de muestras para entrenamiento es menor al 30%. Este valor es significativo, ya que indica que se requiere solo de un 30% de las muestras en entrenamiento para obtener resultados por encima del 80%.

Tal y como ya se había establecido anteriormente, se observa que el método de Canny sigue siendo mucho más óptimo para la obtención de mejores resultados que el método de Otsu, al menos para la base de datos que se utiliza en el presente proyecto.

6.3.3.2 Sistema basado en DCV

En esta sección se presentarán los resultados obtenidos con el sistema que se basa en DCV. Estos resultados se obtienen de forma análoga al sistema anterior basado en PCA, con la diferencia que con DCV el número óptimo de coeficientes son calculados por algoritmo de manera automáticamente mediante métodos

heurísticos. Para el tamaño óptimo de 30x30 píxeles, el número de coeficientes calculado es 4.

En la siguiente tabla se presentan los resultados obtenidos en DCV, de forma similar a como se realizó en el caso del sistema basado en PCA.

Hold-Out (% de test)	Tasa de acierto del sistema basado en DCV	
	Método de Otsu	Método de Canny
50%	73,79%	75,25%
60%	73,28%	75,09%
70%	71,95%	72,50%
80%	64,59%	63,88%
90%	57,20%	58,65%
k-iteraciones	Método de Otsu	Método de Canny
10	76,21%	77,94%
7	74,93%	76,25%
5	73,24%	74,56%
3	72,39%	73,70%

Tabla 9: Resultados obtenidos para DCV

Tal y como se obtuvo con el sistema PCA, con DCV se consigue el mejor resultado (77,94%) con 10k-iteraciones, por las razones explicadas en el apartado anterior acerca del número de muestras usadas para entrenamiento. Sin embargo, los resultados en DCV son insuficientes si se comparan con PCA, por lo que el rendimiento de PCA es bastante más efectivo.

En cuanto a la validación Hold-Out, se puede concluir que a partir del 30% de muestras del conjunto de entrenamiento se pueden obtener tasas de acierto superiores al 70%.

De nuevo al igual que en PCA, el método de Canny globalmente presenta mejores resultados que el método de umbralización de Otsu.

6.3.3.3 Sistema basado en DCT

A continuación, se mostrarán los resultados conseguidos para el sistema basado en DCT, atendiendo a la base de datos preprocesada mediante Otsu y Canny para el centro de la telaraña. La DCT realiza la transformación de cada imagen de forma independiente, a diferencia de PCA o DCV que trabajan sobre el conjunto de las muestras.

Como se hizo en el caso del sistema basado en Análisis de Componentes Principales, antes de mostrar los resultados obtenidos realizando simulaciones con validación cruzada Hold-Out y k-iteraciones, resulta de nuevo interesante mostrar cómo varía la tasa de aciertos del sistema basado en DCT respecto al número de coeficientes utilizados.

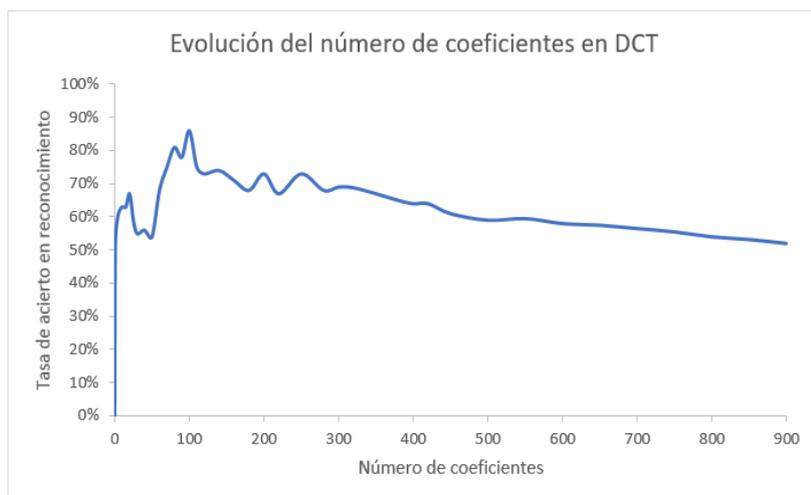


Fig. 30: Evolución del número de coeficientes del sistema DCT

La gráfica superior se ha obtenido a partir de las simulaciones sobre el conjunto de muestras basado en el método de segmentación de detección de bordes de Canny. Se observa que la tasa de acierto es máxima cuando el número de coeficientes utilizado es de 100. A partir de esa cifra empieza a descender, teniendo pequeños máximos locales hasta que se estabiliza entre el 50% y el 60% de tasa de acierto.

En la siguiente gráfica se muestran los resultados para las simulaciones efectuadas, de manera similar a cómo se ha realizado en los métodos anteriores.

Hold-Out (% de test)	Tasa de acierto del sistema basado en DCT	
	Método de Otsu	Método de Canny
50%	81,13%	87,68%
60%	80,66%	85,45%
70%	79,54%	82,61%
80%	73,91%	74,10%
90%	64,15%	66,18%
k-iteraciones	Método de Otsu	Método de Canny
10	84,18%	89,25%
7	82,59%	88,71%
5	81,50%	87,33%
3	80,72%	85,98%

Tabla 10: Resultados obtenidos en DCT

Analizando la tabla de resultados se llega a la misma conclusión que en PCA y DCV en cuanto al método que ofrece la máxima tasa de acierto, así que el sistema se mantiene estable en el método de validación k-iteraciones.

En cuanto a la validación cruzada Hold-Out, en la base de datos preprocesada mediante el método de Otsu, se necesita un mínimo del 40% de las muestras

para entrenamiento si se quiere conseguir resultados superiores al 80%, mientras que en la base de datos preprocesada mediante el método de Canny, basta con un 30% de las muestras de entrenamiento para obtener una tasa de acierto superior al 80%.

En este método, por tanto, también se demuestra que es más adecuada la segmentación basada en la detección de bordes de Canny para el conjunto de imágenes que contiene la base de datos que se utiliza en este proyecto.

6.3.4 Resultados obtenidos con la fusión de sistemas

Como se ha explicado en el Capítulo 4, los sistemas de fusión mejoran los resultados obtenidos mediante los sistemas individuales. En concreto, los sistemas de fusión que se proponen son “el más votado” y “Adding scores”. Puesto que los valores de los vectores de prueba no están correlacionados para PCA, DCV y DCT, se puede estimar que una fusión de los mismos puede arrojar mejores resultados que los sistemas tratados individualmente.

Por otro lado, y dado que los resultados de DCV no presentan una tasa de acierto comparable a la que se obtiene en PCA y DCT, se puede imaginar que la fusión con el sistema DCV no va a suponer una mejora al ser fusionado con los otros dos sistemas, pero de todas formas se presentarán los resultados obtenidos.

En la tabla siguiente se muestran los resultados obtenidos para la fusión de los sistemas, utilizando simulaciones basadas en 10 iteraciones de validación cruzada Hold-Out con el 50% de muestras para entrenamiento.

Método	Tasa de acierto de los sistemas fusionados			
	Método de Otsu		Método de Canny	
	Más votado	Adding scores	Más votado	Adding scores
PCA + DCT	87,33%	87,65%	91,49%	91,62%
PCA + DCT + DCV	87,12%	87,19%	90,86%	91,27%

Tabla 11: Resultados de la fusión de sistemas en identificación

Como puede observarse, el sistema basado en la fusión de PCA y DCT es el que mejor resultado ofrece, para la base de datos segmentada según el método de detección de bordes de Canny, y usando el módulo de fusión “Adding scores”.

Por lo que se ha conseguido mejorar mediante la fusión de sistemas la tasa de acierto obtenida mediante los sistemas individuales.

6.4 Resultados obtenidos en verificación

En tareas de reconocimiento, como se ha visto anteriormente, se calculó que el tamaño óptimo de redimensionamiento es de 30x30 píxeles tanto para el centro de la telaraña como para la tela completa. También se estableció el número óptimo de coeficientes o componentes para cada uno de los sistemas de extracción de características. En tareas de verificación se utilizarán estos parámetros para agilizar el proceso de simulación y obtención de resultados, asumiendo que, tanto en identificación como en verificación, dichos parámetros son óptimos. De igual modo se utilizará la base de datos basada en el método de segmentación de detección de bordes de Canny, ya que es la que mejor resultados ofrecía en identificación. Y concretamente, de esta base de datos se usará el conjunto de imágenes que contiene el centro de la telaraña.

Tal y como se explicó en el capítulo 4.4, el rendimiento del clasificador en verificación se obtiene a partir del área bajo la curva ROC, denominado de forma habitual como AUC.

6.4.1 Resultados obtenidos por los sistemas individuales

Para la realización de simulaciones, se emplearán las mismas técnicas que en identificación. Es decir, se efectuaron 10 iteraciones mediante la técnica de validación cruzada *Hold-Out* y haciendo un barrido sobre el porcentaje para el conjunto de entrenamiento desde el 50% hasta el 10% en pasos de 10. Igualmente, también se realizaron 10 *K*-iteraciones con los siguientes valores de *k*: 3, 5, 7 y 10. De esta manera se garantiza la robustez y fiabilidad del sistema implementado.

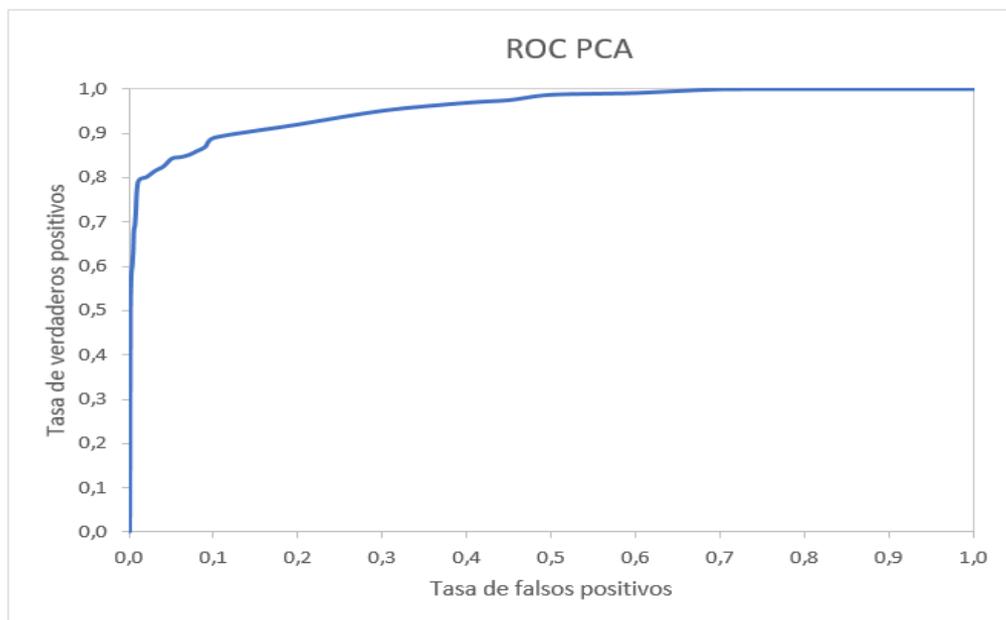


Fig. 31: Curva ROC para el sistema PCA

Como el procedimiento es análogo en los tres sistemas individuales, se van a mostrar los resultados obtenidos en verificación de los tres juntos. Se recuerda que para comparar los distintos sistemas se hará en función al área bajo la curva ROC, o AUC. A modo de ejemplo, se representa en la tabla 12 la curva ROC correspondiente al método PCA para una verificación con validación cruzada Hold-Out con el 50% de muestras de entrenamiento.

En la siguiente tabla se exponen los resultados obtenidos.

Hold-Out (% de test)	AUC (área bajo la curva ROC)		
	PCA	DCV	DCT
50%	90,45%	80,54%	85,70%
60%	89,60%	80,02%	85,17%
70%	88,22%	79,51%	84,33%
80%	87,50%	78,80%	83,16%
90%	87,03%	77,24%	81,75%
k-iteraciones	PCA	DCV	DCT
10	90,97%	81,19%	86,46%
7	90,72%	80,97%	86,28%
5	90,22%	80,99%	86,00%
3	89,85%	80,59%	85,35%

Tabla 12: Resultados obtenidos en los sistemas individuales para verificación

Al igual que en el caso de reconocimiento o identificación, los mejores resultados se obtienen con la técnica de 10k-iteraciones. De nuevo PCA vuelve a ser el método de clasificación con resultados más aceptables. De hecho, al 50% en Hold-Out se puede considerar que el test realizado es muy bueno, ya que supera ligeramente el 90% de AUC. Por otro lado, los test basados en DCV y DCT solo pueden calificarse como buenos, ya que su AUC se encuentra entre el 75% y el 90%.

6.4.2 Resultados obtenidos con la fusión de sistemas

Al igual que en tareas de reconocimiento/identificación, con el fin de optimizar los resultados que se han obtenido con los sistemas individuales, se propone la fusión de los sistemas analizados, también mediante los módulos de fusión “el más votado” y “Adding scores”. Puesto que los valores de los vectores de prueba no están correlacionados para PCA, DCV y DCT, se puede estimar de nuevo que una fusión de los mismos puede arrojar mejores resultados que los sistemas tratados individualmente.

Por otro lado, y dado que los resultados de DCV no presentan tampoco en este caso una tasa de acierto comparable a la que se obtiene en PCA y DCT, se puede imaginar que la fusión con el sistema DCV no va a suponer una mejora al ser fusionado con los otros dos sistemas. Como en tareas de identificación se demostró que el sistema con mayor tasa de acierto provenía de la fusión de PCA

y DCT, en esta sección se mostrarán los resultados en identificación obtenidos para este sistema.

Dado que los resultados no mejoran notablemente los obtenidos en PCA, no se presenta la gráfica ROC correspondiente a la fusión de sistemas, ya que esta es muy similar a la mostrada en la figura 31.

Los valores de AUC obtenidos se muestran en la siguiente tabla:

Hold-Out (% de test)	AUC del sistema basado en PCA + DCV	
	Más votado	Adding scores
50%	90,23%	90,82%
60%	89,57%	90,65%
70%	88,08%	90,04%
80%	87,39%	89,83%
90%	87,08%	88,51%
k-iteraciones	Más votado	Adding scores
10	90,66%	91,96%
7	90,53%	91,83%
5	90,38%	91,64%
3	90,17%	91,31%

Tabla 13: Resultados obtenidos en la fusión de sistemas para verificación

Se observa que, en función de los valores de AUC, mediante el módulo de fusión Adding scores se mejora ligeramente el rendimiento de los sistemas individuales, y con la validación cruzada de 10k-iteraciones se consiguen resultados muy cercanos al 92% de AUC, lo que mantiene el test con una calificación de muy bueno.

6.4 Resumen

Durante el proceso experimental del actual proyecto, se han realizado múltiples simulaciones, entre ellas las que incluyen el cálculo del tamaño óptimo de redimensionamiento de las imágenes. Se ha obtenido que este tamaño es de 30x30 píxeles, para las bases de datos preprocesadas por el método de umbralización de Otsu y por el método de detección de bordes de Canny, tanto para la base de datos de la telaraña completa como la del centro de la telaraña.

De igual manera, se realizaron simulaciones con el fin de determinar el número de coeficientes óptimos de los sistemas de extracción de características implementados.

Una vez comprobado que la tasa de acierto es mayor para la base de datos con las imágenes preprocesadas mediante el método de detección de bordes de Canny y particularmente aquella que contiene las imágenes del centro de la telaraña, se analizan los sistemas individuales para esta base de datos en particular.

En las tareas de reconocimiento o identificación, con el sistema de validación cruzada Hold-Out al 50% de muestras de entrenamiento, se obtuvo el mejor resultado con un 89,24% usando PCA y con un 91,62% de acierto fusionando PCA y DCT [37].

Por otro lado, para las tareas de verificación, se logró una tasa de AUC (área bajo la curva ROC) del 90,97% conseguido con el sistema PCA mediante el método de 10k-iteraciones, y con la fusión de sistemas PCA y DCT basado en PCA se logró una tasa del 91,96%.

Capítulo 7: Conclusiones

7.1 Introducción

En este capítulo se recogerán las principales conclusiones que se pueden extraer del proceso experimental en cada una de las etapas del mismo y de los resultados obtenidos.

Una vez analizados los procesos y explicadas las conclusiones principales que se extraen de los mismos, es adecuado realizar una estimación de las principales líneas de trabajo que se pueden desarrollar tomando como punto de partida este proyecto.

7.2 Conclusiones principales

La conclusión principal que se puede extraer de este proyecto es que a pesar de la dificultad en el tratamiento de la base de datos por la variedad de calidad y contraste de las imágenes (a todas luces insuficiente para realizar un procesamiento de las mismas completamente óptimo), se ha conseguido realizar un sistema de clasificación de especies de arañas aplicando diferentes técnicas de preprocesado. Este sistema realiza tareas tanto de identificación como de verificación, manteniendo porcentaje alto de acierto. Es decir, se han cumplido los objetivos de este proyecto.

Dado que el rendimiento del sistema depende en gran medida de la configuración de la base de datos tratada, se pone de manifiesto la necesidad de cuidar adecuadamente el medio ambiente y las condiciones ambientales (iluminación, contraste, presencia de objetos fuera del material de estudio) al tomar fotografías, así como que son de calidad decente. De esta manera, el preprocesamiento sería más fácil de realizar y la base de datos mucho más óptima para la tarea de reconocimiento.

Otra conclusión interesante que se puede extraer gira en torno a la etapa de preprocesado, y versa sobre el método de segmentación. Hasta ahora, en trabajos previos [10], se había utilizado el método de umbralización de Otsu para binarizar la imagen, de manera que quedara la telaraña como único elemento con píxeles en blanco, dejando el resto de elementos de la imagen en negro.

Para la base de datos con la que se partía, este método presentaba ciertas carencias en la segmentación que provocaban que el método de clasificación no tuviera tan buenos resultados como se esperaba en un principio. Por ello se decidió implementar el método de segmentación basado en la detección de bordes de Canny. A la luz de los resultados obtenidos, se ha demostrado que, para esta base de datos particular, el método de detección de Canny ha sido más exitoso en el reconocimiento y verificación que el método de umbral de Otsu [37].

A la hora de tratar la base de datos ya segmentada, y una vez realizadas las pertinentes simulaciones de extracción de características y clasificación, se llega a dos conclusiones. La primera es que, tal como se planteaba desde un principio, el centro de la telaraña contiene mayor información diferenciadora con respecto a la tela completa, y se traduce en que los resultados obtenidos en la clasificación son mejores para la base de datos que contiene imágenes del centro. La segunda conclusión es que el tamaño de redimensionado que mejor resultados ofrece en la clasificación son de 30x30 píxeles, tanto para la base de datos con la telaraña completa como para la base de datos con el centro de la misma. También este tamaño es el óptimo, tanto para la base de datos preprocesada con el método de Otsu como para la preprocesada con el método de Canny.

En la etapa de extracción de características, se han implementado tres técnicas, PCA (Análisis de Componentes Principales), DCV (Vectores Comunes Discriminantes) y DCT (Transformada discreta del coseno). El método que mejores resultados individuales ha obtenido tanto en identificación como en verificación ha sido PCA, usando un clasificador basado en Máquinas de Vectores Soporte (SVM).

Y, por último, pero no menos importante, se ha demostrado que mediante la fusión de sistemas basados en PCA, DCV y DCT, los resultados individuales de cada uno de los métodos son mejorados, logrando así una tasa de éxito ligeramente superior al 90% [37]. Además, el método de fusión que mejores resultados presenta es el de *adding-scores*.

7.3 Líneas futuras de trabajo

Este proyecto ha trabajado con un método usado para otros trabajos [9] [10], que se ha adecuado a las condiciones específicas del conjunto de imágenes con el que se ha trabajado, y se han planteado métodos en la segmentación y en la clasificación que mejoran los resultados obtenidos para esta misma base de datos con los métodos clásicos. Es por ello que se pueden desarrollar a partir de aquí diferentes estrategias e ideas para trabajar en un futuro que podrían dar más consistencia y robustez a los sistemas planteados. Algunas de estas ideas son:

- Realización de los mismos experimentos del presente trabajo con otras bases de datos con imágenes de mayor calidad y más aptas para el preprocesado. De esta manera se podrá analizar si el método de Canny para la segmentación de las imágenes ofrece de manera general un mejor

resultado a la hora de realizar la clasificación que el método de umbralización de Otsu.

- Desarrollo de otros métodos de extracción de características. Además de PCA, DCV y DCT, se pueden implementar más métodos para analizar si ofrecen mejores prestaciones; por ejemplo, el uso de transformadas Wavelet o análisis discriminantes.
- Podría ser interesante también analizar cómo funcionaría este sistema en concreto si el método de clasificación se basara en métodos bayesianos clásicos o aprendizaje simbólico en lugar de SVM.
- Realizar el presente sistema utilizando otro método de programación, como Java. Con esto se conseguiría una portabilidad del sistema y la utilización del mismo en aplicaciones sin necesidad de recurrir al entorno Matlab.

7.4 Resumen

A la finalización del proyecto, se puede concluir que se ha alcanzado el objetivo principal, que era realizar un sistema de clasificación de especies de arañas aplicando diferentes técnicas de preprocesado que fuera capaz de mantener resultados óptimos en identificación y verificación.

Se proponen nuevas líneas de trabajo a raíz de este PFC, como por ejemplo la realización de experimentos con otras bases de datos para probar el efecto en ellas de la segmentación mediante el método de detección de bordes de Canny, además del estudio de otros métodos de extracción de características y clasificación. Por último, sería interesante en un futuro implementar el sistema en otro lenguaje de programación, para conseguir la portabilidad a otros entornos.

Bibliografía

- [1] Sytnik K.M.: *Preservation of biological diversity: Top-priority tasks of society and state*. Ukrainian Journal of Physical Optics 11(suppl.1), S2-S10 (2010).
- [2] Plataforma NaturaLista - <http://naturalista.conabio.gob.mx/taxa/47118-Araneae> (Última Visita, mayo de 2017)
- [3] Revista electrónica DYNA, sobre ingeniería e Industria - <http://mundozoologia.wordpress.com/2012/05/07/anfibios-anuros/> (Última visita, mayo de 2017)
- [4] Elices Calafat M.: *Las arañas y sus telas, un paradigma disciplinar*. Discurso de ingreso en la Real Academia de Doctores de España, Madrid. 2009
- [5] Gosline J., Guerette P., Ortlepp, Savage K.: *The mechanical design of spider silks: from fibroin sequence to mechanical function*. The Journal of Experimental Biology, 1999: 202.
- [6] Lin L., Sobek W. *Structural hierarchy in spider webs and spiderweb-type systems*. The Structural Engineer, 1998: 76, 59-64.
- [7] Madrigal-Brenes R., Barrantes G.: *Construction and function of the web of Tidarren sisypoides* (Araneae: Theridiidae) The Journal of Arachnology 37: pp. 306–311 (2009).
- [8] Eberhard W.G.: *Behavioral Characters for the Higher Classification of Orbweaving Spiders*. Society for the Study of Evolution. Evolution 36 (5), pp. 1067-1095 (1982).
- [9] Ticay-Rivas J.R., del Pozo-Baños M., Eberhard W.G., Alonso J.B., Travieso González C.M.: *Spider Recognition by Biometric Web Analysis*. IWINAC 2011, Part II, LNCS 6687, pp. 409-417, 2011. Springer-Verlag Berlin Heidelberg 2011
- [10] Gutiérrrez Ramos M.A.: *Reconocimiento de Arañas por medio del análisis biométrico de sus telarañas*. ULPGC 2012.
- [11] Documentación de Matlab R2011a – Mathworks, <http://www.mathworks.com> (Última visita, abril de 2017)

- [12] González R., Woods R.: *Digital Image Processing*. Library of Congress Cataloging-in-Publication Data, pp. 116-137. Second edition (2002)
- [13] Jules R.; Takamura T.: *Alternative Approach for Satellite Cloud Classification: Edge Gradient Application*. Advances in Meteorology. 2013: 1–8.
- [14] Malina W., Ablameyko S., Pawlak W.: *Fundamental Methods of Digital Image Processing*, 2002.
- [15] J. K. Anil.: *Fundamentals of digital image processing*, Prentice Hall, 2004.
- [16] Filtros adaptativos. Filtros de Wiener.
<http://bibing.us.es/proyectos/abreproy/11284/fichero/Volumen+1%252FCap%E Dtulo+2.pdf> (Última vista, mayo de 2017).
- [17] Serra J., Salembier P.: *Mathematical Morphology and its Application to Signal Processing*, proceedings of the 1st International Workshop on Mathematical Morphology and its Applications to Signal Processing,(1993)
- [18] Jolliffe, I.T.: *Principal Component Analysis*, 2nd edn. Springer Series in Statistics (2002).
- [19] Smith LI.: *A tutorial on Principal Components Analysis*.
http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf (Última vista, mayo de 2017).
- [20] Cevikalp H., Neamtu M., Wilkes M., Barkana A.: *Discriminative Common Vectors for Face Recognition*. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, Nº 1, pp. 4 – 13. (2006).
- [21] Diaz-Chito K., Ferri F.J.: *Extensiones del Metodo de Vectores Comunes Discriminantes*, Publicia, Pasta Blanda, 2003.
- [22] Ahmed N., Natarajan T., Rao K.R.: *Discrete Cosine Transform*, IEEE transactions on Computes, pp. 90-93. (1974).
- [23] Oppenheim A, Willsky A., Nawab S.: *Signals and Systems*, Segunda edición. Editorial: Prentice Hall. pp. 358-367, (1998).
- [24] Tjahyadi R., Liu W., Venkatesh S.: *Application of the DCT Energy Histogram for Face Recognition*, Submitted to the Proceedings of the 2nd International Conference on Information Technology for Application, pp. 305-310 (2004).
- [25] Shao X., Johnson S.G.: *Type-II/III DCT/DST algorithms with reduced number of arithmetic operations*. Signal Processing 88(6):1553-1564 · June 2008.
- [26] Cortes C., V. Vapnik V.: *Support-vector networks*. *Machine Learning*, 1995, Kluwer Academic Publishers, Boston. pp. 273-297.

- [27] Surges J.C.: *A tutorial on support vector machines for pattern recognition*. Data Mining and Knowledge Discovery, vol. 2, n° 2, pp. 955-974, (1998).
- [28] Shawe-Taylor J., Cristianini N.: *Support Vector Machines and other kernel-based learning methods*, Cambridge University Press (2000).
- [29] Maldonado S., Weber R.: *A wrapper method for feature selection using support vector machines*. Information Sciences, 179(13):2208_2217, 2009.
- [30] Fawcett T.: *An introduction to ROC analysis*. Pattern Recognition Letters, 27, pp. 861-874. (2006).
- [31] Hanley J.A., McNeil B.J.: *A method of comparing the areas under receiver operating characteristic curves derived from the same cases*. Radiology 148 (3): pp. 839–843. (1983).
- [32] Ludmila I., Kunch E.: *Combining Pattern Classifiers, Methods and Algorithms*, John Wiley & Sons Publications. (2004).
- [33] Ma Y., Cukic B., Harshinder S.: *A Classification Approach to Multimodal Biometric Score Fusion*. Lane Department of Computer Science and Electrical Engineering, Department of Statistics West Virginia University. (2004).
- [34] Joanneum F.H.: *Technical Report. Cross-Validation Explained*, Institute for Genomics and Bioinformatics. (2005-2006).
- [35] Schneider J.: *Technical Report. The holdout method*, the school of computer science, 7 de Febrero de 1997.
- [36] Refaeilzadeh P., Tang L., Lui H.: *k-fold Cross-Validation*, Arizona State University, 6 de noviembre de 2008
- [37] Batista-Plaza D., Travieso Gonzalez C. M, Kishore Dutta M., Singh A. *Biometric analysis for the recognition of spider species according to their webs*. 2017 4th IEEE Uttar Pradesh Section International conference on Electrical, Computer and Electronics (UPCON). IEEE Conference #41590. Mathura (India), 26 Oct - 28 Oct 2017. (DOI: 10.1109/[UPCOM.2017.XXXXXXX](https://doi.org/10.1109/UPCOM.2017.XXXXXXX)) **(Accepted – In press)**

PLIEGO DE CONDICIONES

Pliego de condiciones

Para el presente proyecto, se ha utilizado una serie de herramientas software y hardware, indicadas a continuación.

Como elementos hardware se empleó:

- Un **ordenador portátil** con microprocesador Intel® Core™ i7-7500 a 2,7 GHz, 16 GB de memoria RAM, y 1 TB de disco duro, el cual se utilizó en la fase de desarrollo, entrenamiento y verificación de los sistemas de reconocimiento, para realizar las simulaciones y para la escritura de la memoria del proyecto.

También se utilizaron las siguientes herramientas software:

- **Matlab® versión 7.12.0:** para el desarrollo de la totalidad de los algoritmos implementados en este proyecto. Además del paquete básico se utilizaron los siguientes toolboxes:
 - **Image Processing Toolbox:** para el procesado de las imágenes.
 - **Signal Processing Toolbox:** para el procesado de la señal de audio.
 - **Stadictics Toolbox:** que proporciona funciones y aplicaciones para describir, analizar y modelar datos.
- **Microsoft Office® 2016:** es el paquete de herramientas, en las que se incluye Microsoft Word y Microsoft PowerPoint, que se han utilizado para la elaboración de la memoria y presentación del proyecto.
- **Windows® 10 HOME:** es el sistema operativo bajo el que se trabajó.

Otros elementos software que se utilizaron fueron las bases de datos.

- **Base de datos:**
Nº de imágenes: 185

Nº de clases: 4

Tamaño: véase Capítulo 2.2

Formato de imagen: JPG

PRESUPUESTO

Presupuesto

Don David Batista Plaza, autor del presente Proyecto Fin de Carrera, declara que:

El Proyecto Fin de Carrera con título “Estudio de las técnicas de procesado de imágenes para la clasificación de especies de arañas”, desarrollado en la Escuela de Ingeniería de Telecomunicación y Electrónica en la Universidad de Las Palmas de Gran Canaria, en el tiempo de seis meses, tiene un coste de desarrollo total de **31.855,66 €**, correspondiente a la suma de las cantidades consignadas a los apartados considerados a continuación.

El autor de proyecto
David Batista Plaza

Las Palmas de Gran Canaria a 21 de Julio de 2017.

P.1 Desglose del Presupuesto

Para la realización del presupuesto se han seguido las recomendaciones del Colegio oficial de Ingenieros de Telecomunicación (COIT) y la Asociación Española de Ingenieros de Telecomunicación. El presupuesto se ha desglosado en varias secciones en las que se han separado los distintos costes asociados al desarrollo del proyecto. Estos costes se dividen en:

1. Recursos materiales.
2. Trabajo tarifado por tiempo empleado.
3. Costes de redacción del proyecto.
4. Material fungible.
5. Derechos de visado del COIT.
6. Gastos de tramitación y envío.
7. Aplicación de impuestos.

P.2 Recursos Materiales

Entre los recursos materiales utilizados para la realización de este proyecto se incluyen las herramientas software de desarrollo de los algoritmos del sistema, los paquetes software usados para la redacción de la memoria, y el sistema operativo bajo el que se ejecutó el trabajo. Asimismo, se incluyen los equipos hardware usados para dar soporte a estas herramientas.

Se estipula el coste de amortización para un período de 3 años. Para ello, se utilizará un sistema de amortización lineal o constante, en el que se supone que el inmovilizado material se deprecia de forma constante a lo largo de su vida útil. La cuota de amortización anual se calcula usando la siguiente fórmula:

$$\text{Cuota Anual} = \frac{\text{Valor de adquisición} - \text{Valor residual}}{\text{Número de años de vida útil}}$$

El “valor residual” es el valor teórico que se supone que tendrá el elemento después de su vida útil, teniendo en cuenta los índices de depreciación actual.

En el caso de hardware y software son 3 años (al 33% de depreciación máximo por año).

P.2.1 Recursos software

Las herramientas software utilizadas en presente proyecto fueron:

- Matlab® v.7.12.0
- Microsoft Office® 2016
- Windows® 10 HOME

Teniendo en cuenta que la duración del proyecto es de aproximadamente de un año y el cálculo del coste de amortización se establece en un período de 3 años, los costes de amortización se calcularán para los seis primeros meses. Estos costes se pueden ver en la tabla de costes de las herramientas software.

Costes de las herramientas Software			
Descripción	Coste Total	Tiempo de uso	Valor Amortización
Matlab® v.7.12.0	3900 €	6 meses	650 €
Microsoft Office® 2016	350 €	6 meses	58,33 €
Windows® 10 HOME	300 €	0 €	50 €
Total de Costes			758,33 €

Tabla 14: Costes de los recursos software

Por lo tanto, el coste total del material software libre de impuestos asciende a *setecientos cincuenta y ocho euros con treinta y tres céntimos de euro (758,33 €)*.

P.2.2 Recursos Hardware

Las herramientas hardware en la que se apoya el presente proyecto son:

- Un **ordenador portátil** con microprocesador Intel® Core™ i7-7500 a 2,7 GHz, 16 GB de memoria RAM, y 1 TB de disco duro.

Aplicando la regla de costes anterior se tiene:

Costes de las herramientas Hardware			
Descripción	Coste Total	Tiempo de uso	Valor Amortización (1 año)
Ordenador portátil	1200 €	6 meses	200 €
Total de Costes			200 €

Tabla 15: Costes de las herramientas Hardware

Por lo tanto, el coste total del material hardware libre de impuestos asciende a *doscientos* (**200 €**).

P.3 Trabajo Tarifado por Tiempo Empleado

El proyectante ha invertido 6 meses en las tareas de formación, especificación, desarrollo y documentación necesarias para la elaboración del presente proyecto fin de carrera. A continuación, se contabilizan los gastos correspondientes a la mano de obra. Para su cálculo se ha utilizado la última fórmula del COITT, que en su día se conocía como “Propuesta de Baremos Orientativos para el Cálculo de Honorarios”. Hay que comentar que, en la actualidad, todos los colegios profesionales han recibido una nota del Ministerio de Economía y Hacienda en la que se recuerda que, siguiendo directivas europeas, se deben eliminar los baremos orientativos de los honorarios que tradicionalmente se venían aplicando. En esta nota informativa se indica que, los honorarios son libres y responden al libre acuerdo entre el profesional y el cliente.

La fórmula propuesta por el COITT (antes de la nota) es:

$$H = (C \times 75 \times Hn) + (C \times 95 \times He)$$

Donde:

H son los honorarios totales por el tiempo dedicado.

Hn son las horas normales trabajadas (dentro de la jornada laboral)

He son las horas especiales.

C es un factor de corrección función del número de horas trabajadas.

Teniendo en cuenta que el tiempo destinado para la formación y documentación no se incluye en el cálculo debido a que estas tareas son necesarias y benefician sólo al alumno, para la realización de este proyecto han sido necesarias 800 horas (8 horas/día · 5 días/semana · 4 semanas/mes · 5 meses), todas ellas dentro del horario normal.

Según el COIT, el coeficiente **C** tiene un valor variable en función del número de horas empleadas de acuerdo con la siguiente tabla:

Horas empleadas	Factor de corrección C
Hasta 36 horas	1,00
De 36 a 72 horas	0,90
De 72 a 108 horas	0,80
De 108 a 144 horas	0,70
De 144 a 180 horas	0,65
De 180 a 360 horas	0,60
De 360 a 540 horas	0,55
De 540 a 720 horas	0,50
De 720 a 1080 horas	0,45
Más de 1080 horas	0,40

Tabla 16: Factor de corrección en función del número de horas invertidas

Al ser el número de horas comprendido entre 720 y 1080 el parámetro C se establece en 0,45, por lo que se tiene que:

$$H = (0,45 \times 75 \times 800) + (0,45 \times 95 \times 0) = 27.000 \text{ €}$$

En la tabla P.4 se desglosa el tiempo de trabajo invertido.

Costes por Tiempo Empleado			
Descripción	Tiempo	Coste/mes	Importe
Formación	0,5 mes	0 €	0 €
Documentación	0,5 mes	0 €	0 €
Especificación	1 meses	5.400 €	5.400 €
Desarrollo	4 meses	5.400 €	21.600 €
Total de Costes			27.000 €

Tabla 17: Costes por Tiempo Empleado

Los honorarios totales por tiempo dedicado libres de impuestos ascienden a *veintisiete mil euros (27.000 €)*.

P.4 Costes de redacción del proyecto

Haciendo uso del punto 1 de la recomendación del COITT, que hace referencia al valor monetario de la redacción de proyectos y trabajos en general, se aplicará la fórmula siguiente para determinar el coste asociado a la redacción de la memoria de este proyecto:

$$R = 0,05 \times P$$

Donde:

R es el coste de redacción.

P es el presupuesto del proyecto.

En la siguiente tabla se muestra el presupuesto calculado hasta el momento, más los costes de materiales fungibles (que se explican en la sección P.5):

Descripción	Costes
Recursos Software	758,33 €
Recursos Hardware	200 €
Trabajo Tarifado por Tiempo Empleado	27.000 €
Material fungible	190 €
Total de Costes	28.148,33 €

Tabla 18: Presupuesto de ejecución material

De este modo, se obtiene que:

$$R = 0,05 \times 28.148,33 = 1.407,42 \text{ €}$$

Por tanto, el coste libre de impuestos derivado de la redacción del proyecto es de *mil cuatrocientos siete euros con cuarenta y dos céntimos de euro* (1.407,42 €).

P.5 Material fungible

Además de los recursos hardware y software, en este proyecto se han empleado otros materiales, como son los folios y el tóner de la impresora entre otros, que se especifican como material fungible. En la tabla P.6. se muestran los costes generados por estos recursos.

Descripción	Costes
Folios	20 €
Tóner de la impresora	140 €
Encuadernación	30 €
Total de Costes	190 €

Tabla 19: Coste del material fungible

P.6 Derechos de visado del COIT

El COITT establece para la redacción de proyectos y trabajos en general los derechos de visado, que, a partir del 1 de marzo del 2007 se calculan de acuerdo con la siguiente ecuación:

$$V = 0,007 \times P \times C$$

Donde:

V es el coste del visado.

P es el presupuesto del proyecto.

C es el coeficiente reductor en función del presupuesto del proyecto.

El presupuesto **P** calculado hasta el momento asciende a la suma de los costes de ejecución material, de redacción y de material fungible. El presupuesto hasta ahora viene dado por:

Descripción	Costes
Coste del tiempo empleado en la ejecución	28,148,33 €
Redacción del proyecto	1.407,42 €
Total de Costes	29.555,75 €

Tabla 20: Valor del presupuesto para el cálculo del visado

Como el coeficiente **C** para presupuestos de menos de 30.050 €, viene definido por el COIT con un valor de 1, el coste de los derechos de visado del proyecto asciende a la cantidad de:

$$V = 0,007 \times 29.555,75 \times 1 = 206,89 \text{ €}$$

Por tanto, el coste de los derechos de visado del proyecto asciende a *doscientos seis euros con ochenta y nueve céntimos de euro (206,89 €)*.

P.7 Gastos de tramitación y envío

Los gastos de tramitación y envío según la tarifa ascienden a *nueve euros* por cada documento visado digital (**9 €**).

P.8 Aplicación de impuestos

El coste total del proyecto, antes de aplicarle los correspondientes impuestos, asciende a *veintinueve mil setecientos setenta y un euros con sesenta y cuatro céntimos de euro (29.771,64 €)*, tal y como se ha calculado en la siguiente tabla:

Descripción	Costes
Presupuesto base	29,555,75 €
Derechos de visado	206,89 €
Gastos de tramitación y envío	9 €
Total de Costes	29.771,64 €

Tabla 21: Presupuesto total sin impuestos

A esto hay que sumarle el 7% de IGIC del presupuesto total sin impuestos, con lo que el coste definitivo del proyecto es:

Costes Totales del Proyecto		
Descripción	Coste Parcial	Total
Recursos Materiales		958,33 €
Software	758,33 €	
Hardware	200 €	
Coste de Ingeniería		27.000 €
Coste de Redacción		1.407,42 €
Material Fungible		190,00 €
Derechos de Visado		206,89 €
Tramitación y Envío		9 €
Subtotal:		29.771,64 €
Aplicación de Impuestos (7% I.G.I.C)		2.084,02 €
Total de Costes		31.855,66 €

Tabla 22: Costes Totales del Proyecto

El presupuesto total asciende a la cantidad de *TREINTA Y UN MIL OCHOCIENTOS CINCUENTA Y CINCO EUROS CON SESENTA Y SEIS CÉNTIMOS DE EURO (31.855,66 €)*.

Las Palmas de Gran Canaria, a 21 de julio de 2017.

Fdo.: David Batista Plaza.

ANEXO A

A.1 Introducción

Junto con esta memoria se adjunta un CD que recoge el trabajo realizado a lo largo de este proyecto. El contenido de este CD es el siguiente: la memoria en formato PDF, el código de las funciones que han sido implementadas en Matlab y la base de datos utilizada.

A.2 Descripción del contenido

Al introducir el CD adjunto en esta memoria se observarán una serie de carpetas:

- Memoria.
- Código.
- Bases de datos.

En la carpeta llamada *Memoria*, se encuentra un archivo PDF el cual se corresponde con la redacción de la memoria. La memoria está compuesta por 7 capítulos y dos anexos. La carpeta *Código* contiene cada una de las funciones utilizadas en este proyecto realizadas en Matlab. Por último, la carpeta *Bases de datos* contiene en tres subcarpetas: la base de datos original de la que se partió, la base de datos recortada con la información de la telaraña y la nueva base de datos que contiene las imágenes para el preprocesado de Otsu y Canny, tanto del centro de la telaraña como de la telaraña completa.

ANEXO B

B.1 Descripción del contenido

En este segundo anexo se encuentra el artículo *Biometric analysis for the recognition of spider species according to their webs*, presentado y aceptado para su publicación en el “2017 4th IEEE Uttar Pradesh Section International conference on Electrical, Computer and Electronics (UPCON)”

Se añade también a continuación la carta de aceptación para la presentación de dicho artículo para “IEEE Conference #41590” en Mathura (India) del 26 al 28 de octubre de 2017.

De: UPCON2017 [mailto:upcon2017@easychair.org]
Enviado el: miércoles, 28 de junio de 2017 3:15
Para: Carlos M Travieso Gonzalez <ctravieso@dsc.ulpgc.es>
Asunto: 4th IEEE UPCON 2017-Registration of Accepted Paper 327-(Do the early bird registration to get the discount).

Dear David Batista-Plaza, Carlos M Travieso Gonzalez, Malay Kishore Dutta and Anushikha Singh,

Greetings from UPCON2017 program committee and technical committee,

Register each your accepted paper to the conference by following instructions of registration available at conference website.
<http://www.gla.ac.in/upcon2017/req.html>

*****Early bird Registration deadline: 05 July, 2017 ***

If more than one author/s of same paper wishes to attend the conference to present their paper, then one has to do full registration and others have to register as co author i.e. 60% of applicable registration fee. Only Full-Time Students are considered under Student's Registration. If an author is Register as Student then he/she must show/send a copy of valid Student ID

Note1: You will receive the copyright form, Camera ready paper submission guidelines with IEEE Template through separate email by us up to July 15, 2017.

Presently all the accepted papers to the conference are required to register first (Do the early bird registration to get the discount). Please email to us filled registration form with registration fee information at: req.upcon2017@gmail.com.

Note2: If you have done already done the registration of your accepted papers, kindly ignore this email

For any query related to registration feel free to write us on req.upcon2017@gmail.com.

Seeking to welcome you during IEEE UPCON2017 and Birth Place of Lord Krishna!!

best regards,

TPC:UPCON2017

Biometric analysis for the recognition of spider species according to their webs

David Batista-Plaza, Carlos M. Travieso
Signals and Communications Department
University of Las Palmas de Gran Canaria
Las Palmas de Gran Canaria, Spain
carlos.travieso@ulpgc.es

Malay K. Dutta
Amity University
Noida, India
mkdutta@amity.edu

Abstract—This work presents a biometric approach for spider identification based on transform domain and Support Vector Machines as classifier. The dataset is composed by 185 images of spider web. The goal of this work is to use the structure of spider web for identifying the kind of spider. The experiments were done using two different of segmentation blocks and the analysis of the whole and center of the spider web. The best accuracy is reached after to run the different combinations.

Keywords—image processing, image segmentation, pattern recognition; classification system

I. INTRODUCTION

Conserving and protecting the biodiversity of our planet is one of the most important fields of work of the scientific community [1]. In the present case, we will study the species of chelated arthropods of the order Araneae, commonly known as spiders. In particular, the object of study will be the spider web.

In the last years, several works have studied the behavior of the spiders and the construction of their webs. The article *Behavioral Characters for the Higher Classification of Orbweaving Spiders* [2], describes the behavior of the spiders during the construction of the spider webs. Each spider subspecies may have a specific pattern of web construction, so it is interesting to study it as a method of recognizing this subspecies.

In *Computer Simulation of Orb-Web Construction* [3], it is detailed that the spider web is a form of record on the behavior of the spider. The spider *Uloborus diversus* adjusts its behavior while laying sticky spiral in response to stimuli it receives as it moves. The course of the last loop of sticky spiral determines the path the spider takes. Spiders, probably aware of the angles that form the sticky spiral they make, use this information in making decisions to turn back. Computer-simulation has shown that a simple turnback program using just these angles can produce a spider-like pattern of spiral.

In the publication *Spider Recognition by Biometric Web Analysis* [4] the authors propose a series of techniques to carry out this task. These techniques are based on biometric analysis, such as image processing tools, Main Component Analysis and Support Vector Machines to build a spider web identification system, using a database made with spider web images from three species of spiders.

The present paper describes a new system of recognition of four species of spiders from their webs. The images were taken in different work environments and do not have the characteristics of contrast and quality necessary for the study of a biometric classification system, so it will be necessary to perform a preprocessing of the image. Subsequently, the characteristics of each group of images should be studied. Next, classification should be done using different biometric methods. Once the classification is done, the results should be extracted.

Spiders are very fragile species and the variation of their population is an indicator of the destruction of biodiversity. Spiders are not always on their webs, making it easier to use webs for identification. Starting from this, the present paper develops a tool that facilitates the work of biologists and at the same time contributes to the conservation of biodiversity by providing knowledge. It seeks to achieve these goals by presenting an innovative design that uses known and new methods and combines them to build an efficient system.

In Section II, the characteristics of the database are provided. Methods and algorithms used are explained in section III. In section IV experimental methodology is presented. In section V experimental results are presented and discussed. Conclusion and future work is given in section VI.

II. DATABASE

The database consists of a set of web images of four different species. Three of them belong to the Araneidae type: *Araneus diadematus*, *Eustala illicita* and *Cyclosa caroli*. The fourth belongs to the type Nephilidae and its scientific name is *Nephila edulis*. The following table details the number of images of each species:

TABLE I. DATASET OF SPECIES OF SPIDERS

Species	Number of images
<i>Eustala illicita</i> (Araneidae)	76
<i>Cyclosa caroli</i> (Araneidae)	38
<i>Nephila edulis</i> (Nephilidae)	33
<i>Araneus diadematus</i> (Araneidae)	38

Of the species *Eustala illicita*, although the images were obtained in a controlled environment in laboratory and in grayscale, the illumination and the contrast of the web are not adequate.

The images of the species *Cyclosa caroli* have also been taken in grayscale, but with better illumination and better isolation. However, the quality is lower than *Eustala illicita* images.

The third species *Nephila edulis* has the most careful images in terms of quality and isolation of the spider web in the laboratory, and they are the most acceptable for optimal preprocessing.

The fourth and last, *Araneus diadematus*, presents images with quality characteristics similar to *Nephila edulis*. The dark and uniform background facilitates the spider web to have a better contrast, also good for an optimal pre-processing. However, the quality in about 30% of the images is not quite good.

Samples of images of each species are shown in Fig 1.

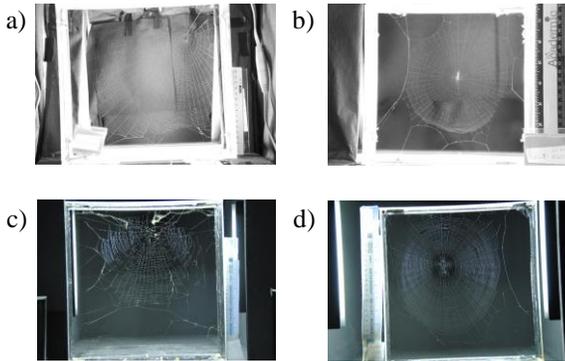


Fig. 1. Examples of spider web images. a) *Eustala illicita* b) *Cyclosa caroli* c) *Nephila edulis* d) *Araneus diadematus*

III. METHODS

The methods will be divided into three sections. In the first one the methods of pre-processing will be studied, in the second the methods of extraction of characteristics and in the third the systems of classification.

A. Preprocessing

To perform the preprocessing, several techniques have been used, which will be analyzed below and emphasized in Otsu and Canny methods.

1) *Spatial filtering*: This operation [5] is applied to an image to highlight or attenuate spatial details in order to improve visual interpretation or facilitate further processing. Space operations consist of a processing of the form:

$$v = T(u, u_1, u_2, \dots, u_{p2-1}) \quad (1)$$

Where u and v are the original and transformed values respectively of a certain pixel, and $(u_1, u_2, \dots, u_{p2-1})$ are the intensity values of the neighboring pixels of the pixel under study. It is common practice to represent these operators using a value mask, which consists of a grid of $P \times P$ pixels, typically with odd P and centered around the pixel under study.

For the spatial filtering, a low-pass filter can be used. In this kind of filter, the mask coefficients are all positive. Usually, the coefficients are normalized and their sum is equal to one. Thus, the gain of the filter for null spatial frequencies is unitary, so that the continuous component of the image is not altered. This type of filtering is intended to reduce acquisition noise when only a single image is available. The disadvantage of this technique is the smoothing of the transitions which makes the final appearance of the image to be of a certain blur if the filter is large.

2) *Otsu thresholding*: Thresholding techniques [6] are image segmentations techniques based on image-space regions. The fundamental principle of thresholding techniques is based on the characteristics of the image. In the practice, thresholding creates binary images from grey level ones by turning all pixels below some threshold to zero and all pixels above that threshold to one.

$$g(x, y) = \begin{cases} 0, & f(x, y) < \text{threshold} \\ 1, & f(x, y) \geq \text{threshold} \end{cases} \quad (2)$$

In Otsu's method we search for the threshold that minimizes the intra-class variance, defined as a weighted sum of variances of the two classes:

$$\sigma_w^2(t) = \omega_0(t)\sigma_0^2(t) + \omega_1(t)\sigma_1^2(t) \quad (3)$$

Weights w_0 and w_1 are the probabilities of the two classes and σ_0^2, σ_1^2 are variances of these two classes. The class probabilities are computed from the L histograms:

$$\omega_0(t) = \sum_{i=0}^{t-1} p(i) \quad (4)$$

$$\omega_1(t) = \sum_{i=t}^{L-1} p(i) \quad (5)$$

Otsu shows that minimizing the intra-class variance is the same as maximizing inter-class variance [7]:

$$\begin{aligned} \sigma_b^2(t) &= \sigma^2 - \sigma_w^2(t) \\ &= \omega_0(\mu_0 - \mu_T)^2 + \omega_1(\mu_1 - \mu_T)^2 \\ &= \omega_0(t)\omega_1(t)[\mu_0(t) - \mu_1(t)]^2 \end{aligned} \quad (6)$$

Which is expressed in terms of class probabilities w and class means μ :

$$\mu_0(t) = \sum_{i=0}^{t-1} i \frac{p(i)}{\omega_0} \quad (7)$$

$$\mu_1(t) = \sum_{i=t}^{L-1} i \frac{p(i)}{\omega_1} \quad (8)$$

$$\mu_T = \sum_{i=0}^{L-1} i p(i) \quad (9)$$

The following relations can be easily verified:

$$\omega_0 \mu_0 + \omega_1 \mu_1 = \mu_T \quad (10)$$

$$\omega_0 + \omega_1 = 1 \quad (11)$$

The class probabilities and class means can be computed iteratively. This idea yields an effective algorithm in which it computes histogram and probabilities of each intensity levels. After that, set up initial $w_1(0)$ y $\mu_1(0)$ and step through all possible thresholds and update the previous values and compute the variances. Finally, desired threshold corresponds to the maximum variance.

3) *Canny edge detection*: The Canny edge detector is regarded as one of the best edge detectors currently in use [8]. Canny's edge detector ensures good noise immunity and at the same time detects true edge points with minimum error. Its main steps are:

1. A convolution is performed with a Gaussian filter and the image is smoothed.

2. The gradient of the smoothed image is calculated to determine the pixels where maximum variation occurs (largest modulus of the gradient vector). The direction of the gradient vector is also determined.

3. The matrix M corresponding to the gradient module of the Gaussian function will have large values where the intensity variation is large. Those pixels that are not local maximum in the direction of the gradient are eliminated.

4. A double thresholding process is performed to determine the edge pixels. The pixels with value above a threshold T1 are marked. Those pixels connected to the first ones whose value is above a second threshold T2 ($T2 < T1$) are marked. This will eliminate false edges or double edges.

B. Feature Extraction

Feature extraction starts from an initial set of measured data and builds derived values intended to be informative and non-redundant. Feature extraction, besides being able to reduce the dimensionality of the data, can improve the rate of

generalization and decrease the computational requirements of the classifier. Therefore, the following sections will explain the different methods that have been implemented in order to extract the information contained in the images. The most relevant are:

1) *Principal component analysis (PCA)*: This technique [9] has as objective that given n observations of p variables, it is analyzed if it is possible to adequately represent this information with a smaller number of variables constructed as linear combinations of the originals. To calculate the main components we have to follow these steps:

1. The data must be arranged in matrix form (the rows being the samples and the columns the variables), subtracting the mean for each dimension and calculating the values and eigenvectors of the covariance matrix.

2. The components are chosen and a base is formed with its vectors. Then, the system orders the λ_i eigenvalues and start choosing for the first one until the sum of the remainder is equal to or less than λ_i .

3. Understanding that the sum total of eigenvalues is an expression of the variance of the data, a percentage (80%-90%) of cut can be used that allows to discard those dispersion directions that do not contribute anything or almost nothing of information.

2) *Discrete cosine transformation*: The DCT [10] is a transform similar to the discrete Fourier transform unlike that the sine coefficients are eliminated, so it is an actual transform, because the base vectors, are composed exclusively of cosine functions sampled. The idea of using DCT for pattern recognition is to calculate the DCT and retain a subset of the coefficients that will make up the characteristic vector describing the image space. Given an input sequence $u(n)$, the DCT obtained by the equation:

$$v(k) = \alpha(k) \sum_{n=0}^{N-1} u(n) \cos\left(\frac{(2n+1)\pi k}{2N}\right) \quad 0 \leq k \leq N-1 \quad (12)$$

Where:

$$\alpha(0) = \sqrt{\frac{1}{N}}, \quad \alpha(k) = \sqrt{\frac{2}{N}} \quad 1 \leq k \leq N-1 \quad (13)$$

This means that by applying the DCT to an input sequence, it simply decomposes into a heavy sum of basic cosine sequences.

C. Classification techniques

The classification technique used is based on Support Vector Machine (SVM). SVM [11] are supervised learning

models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier.

More formally, a SVM constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier.

The SVM used have two basic modes of operation:

1) *Training mode*: The system needs to be trained to create a model that will later be used in the test or classification phase. To create this model, the system receives as inputs a series of patterns which must output a known result. This result is introduced by tags, which indicate to which class each sample belongs. This type of training is called supervised training. A model will be generated for each SVM used.

2) *Classification mode*: When an unknown sample is introduced, the classification system should be able to decide which of the classes under study belongs to the sample. The system will use the models generated in the phase of training, analyzing them and determining for each SVM to which class it belongs. In the case where the classification of positive class is given in more than one classifier, the assigned class will be the one with the highest margin.

Several SVMs will be used, which will allow to establish the differences between the different classes using the strategy of one against the rest, where each class is confronted with the other classes acting as if they were a single class.

IV. EXPERIMENTAL METHODOLOGY

In this part, the different proposals of systems for the classification of the spider webs of the four species are presented. The purpose of this chapter is to know the implementation of the contents explained in the previous chapters. The experimental methodology used will be structured into several steps.

A. Processing of the database

The database must be prepared to perform a correct feature extraction. Pursuing this, we preprocess the database with the explained methods. As a result, we will have two segmented image databases, one using the Otsu method and another using the Canny method.

Studying the collection of the images of the webs, it can be observed that in the center of them is where the images differ most. Therefore, we can determine that the center of the web

contains most of the information. So, it is interesting to build a new database for each method, cropping the images to isolate the center. Finally, we have a database with center and full web images for Otsu method and another one with center and full web images for Canny method.

To be able to analyze images automatically, they are resized to have a certain width w and high h . Then the rows or columns of the respective images are concatenated to form a vector of dimension $d = w \times h$, usually of large dimensions. The vector space resulting from the vectorization of the image is called an image space.

The images are resized to several sizes (from 10x10 to 50x50 pixels, following a progression of ten in ten) to assure the independence of the experiments from the image acquisition processing and to find the optimal resizing length.

B. Data validation

In experimentation, we want to measure the performance achieved by a learning algorithm. The data set is divided into two subsets: one of training through which system learns the patterns to be classified and another subset to test the system through the ability to predict against an unknown data set.

A common rule is to use 50% of the database for training and the remaining 50% for tests. The division of the database into test and training subsets is usually done randomly to ensure that the samples are random from the same distribution. As an example, this would be the following distribution:

TABLE II. DISTRIBUTION OF DATASET FOR TRAINING AND TEST

Subset	<i>Eustala ilicita</i> (76 images)	<i>Cyclosa caroli</i> (38 images)	<i>Nephila edulis</i> (33 images)	<i>Araneus diadematus</i> (38 images)
Training	38	19	17	19
Test	38	19	16	19

To generate this distribution, we will use cross-validation techniques, especially the Hold-out validation method. With this method, the data set is divided into two subsets so that the samples do not overlap (they are not repeated in both subsets). This division is performed according to the approximation of the percentage selected for the training set, 50% in this case.

A single cross-validation of the samples is inconclusive because different results can be obtained depending on how the samples are distributed for testing and training. Therefore, we will perform 10 iterations using the Hold-out method.

C. Feature extraction

Once the samples of the database have been correctly distributed between training and test samples, we proceed to feature extraction by the previously explained methods.

In PCA method, the number of principal components is given by the size of the image space. For example, in a 10x10 image, the maximum number of principal components we can calculate is 100. The appropriate number will be chosen by

performing a skipping analysis on the number of components, in order to select the one that gives the best results. And for the present system, the best result has been 10 components.

In DCT method, through the discrete cosine transform, each transformation is performed for each sample or image independently of the rest. By applying this transform, a reduction of the information is achieved by using the first coefficients obtained, which represent the highest energy values. To determine the number of optimal coefficients of the DCT for the system, a skipping analysis will be performed again. In this case, the optimal result has been 100 coefficients.

D. Classification

As explained above, once the feature extraction has been performed, the samples will be sorted by SVM.

We will use "one versus all" strategy. In this strategy, each class is confronted with others, who act as if they were a single class. That is, the SVM classifier will decide whether a sample is of that class or not. Therefore, in this case, the SVM number matches the number of classes. Considering an example with four classes (A, B, C and D), four SVMs will be used, with the following distribution: A versus the rest (B, C and D), B versus the rest (A, C and D), The C in front of the rest (A, B and D) and the D in front of the rest (A, B and C). Using this strategy in our system, four models will be generated.

Each feature test vector must be analyzed by the four models. For each model, we will obtain an output value, which provides information about the similarity of the feature vector that is studied with respect to each of the classes. The maximum value will correspond to the class with which the most similarity obtains. Using a confusion matrix, we will compare the results obtained in each sample with the class to which it actually belongs, in order to calculate the success rate.

Finally, to improve the results obtained, we will make a fusion of the two feature extraction methods.

V. EXPERIMENTAL RESULTS

In this chapter, the results obtained in each process of the experiment are presented, dividing them into several sections.

A. Image segmentation results

The Otsu and Canny methods yield slightly different results in some samples but clearly differentiated in others. For example, taking a sample of the species *Nephila edulis* (whose image has been cut before the pre-processing to isolate the web), we obtained the following results:

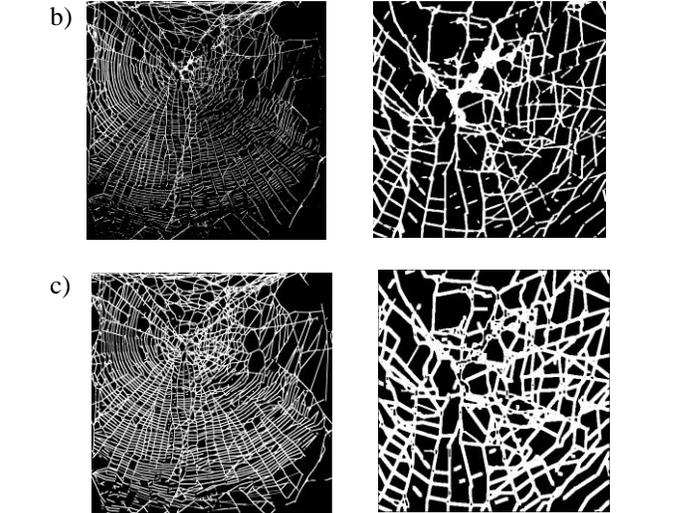


Fig. 2. Examples of segmentation. a) Original images b) Otsu segmented images c) Canny segmented images

However, not all samples have the same lighting and quality conditions. For example, if we take a sample of poor quality of the species *Araneus diadematus*, the results are as follows:

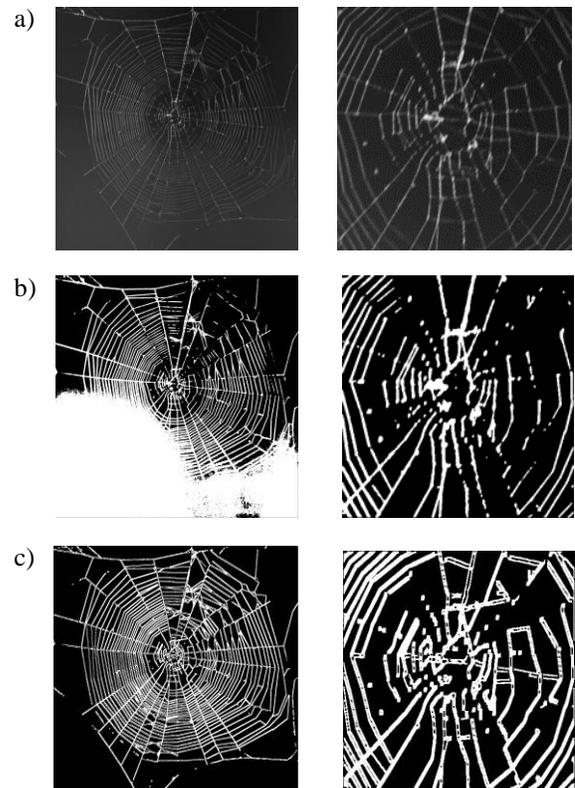


Fig. 3. Results of segmentation for same lighting and quality conditions, according to Fig. 2.

As we can see, the poor quality of images causes that in Otsu method, the system detects as white enough pixels that should not be white, which can increase the error rate. Therefore, Canny method seems to yield better visual results.

A. Classification results

First of all, several simulations were carried out to calculate the optimal resizing size of the web, using half of samples for training and the other half for testing. The results that will be shown correspond to the center of the web, because this zone contains the major part of the information, both for Otsu and Canny methods.

TABLE III. RESULTS OF THE DIFFERENT EXPERIMENTS

System	Maximum success rate (by resize length)				
	10x10	20x20	30x30	40x40	50x50
PCA (Otsu)	82,8%	83,4%	84,2%	80,5%	77,5%
PCA (Canny)	87,3%	84,6%	89,2%	81,5%	78,1%
DCT (Otsu)	77,5%	79,6%	81,1%	80,4%	74,8%
DCT (Canny)	84,1%	82,5%	87,7%	80,8%	75%

As we can see, the best hit rates are given for the Canny method and size 30x30. We have to remember that these results are given considering that for PCA the optimal number of components is 10 and for DCT the optimal number of coefficients is 100.

Now, the next two graphs analyze the success rate for the evolution of the number of main components in PCA and of coefficients in DCT, for a size of 30x30.

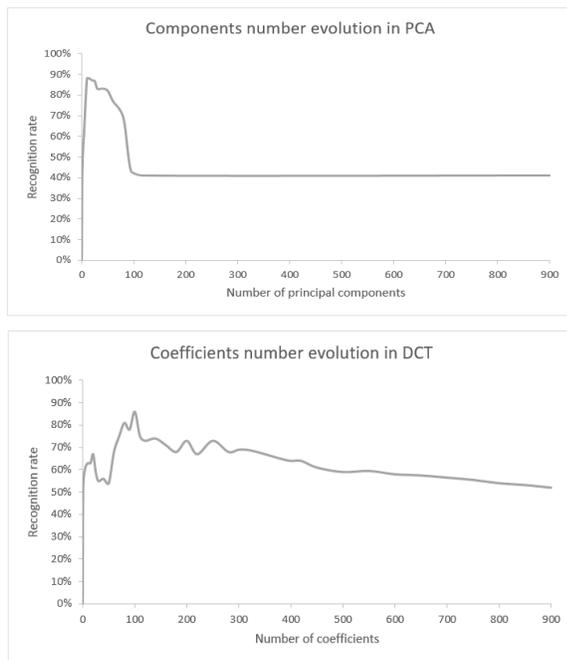


Fig. 4. Results for transform domains

In order to improve the results, system PCA and system DCT will be merged. Since the values of the test vectors are not correlated for PCA and DCT, we can merge the systems using the "adding scores" method, which consist of adding the vectors with the output values provided by the SVM in test mode for a given sample to later apply the rule of the

maximum, where the given class is the one that obtains the highest score or value. The result for this method (with image size 30x30) is presented in the following table:

TABLE IV. THE BEST ACCURACIES OF THE EXPERIMENTS

System	Maximum success rate (size 30x30)
	Adding scores PCA + DCT
Otsu	87,65%
Canny	91,62%

VI. CONCLUSION

Several conclusions can be drawn from this document. The first of these is the need to take proper care of the environment and environmental conditions (lighting, contrast, presence of objects outside the study material) when taking photographs, as well as that they are of decent quality. In this way, the pre-processing would be easier to perform and the database much more optimal for the recognition task.

A second interesting conclusion is about the segmentation method. It has been shown that for this particular database, the Canny edge detection method has been more successful in recognition than the Otsu thresholding method.

And last but not least, it has been proven that by merging systems based on PCA and DCT, the individual results of each of the methods are improved, thus achieving a success rate of slightly higher than 90%.

REFERENCES

- [1] Sytnik, K.M.: "Preservation of biological diversity: Top-priority tasks of society and state". Ukrainian Journal of Physical Optics 11(suppl.1), S2-S10 (2010).
- [2] Eberhard, W.G.: "Behavioral Characters for the Higher Classification of Orbweaving Spiders". Society for the Study of Evolution. Evolution 36 (5), pp. 1067-1095 (1982). K. Elissa, "Title of paper if known," unpublished.
- [3] Eberhard, W.G.: "Computer Simulation of Orb-Web Construction". J. American Zoologist, pp. 229-238 (February 1, 1969).
- [4] Jaime R. Ticay-Rivas, Marcos del Pozo-Baños, William G. Eberhard, Jesús B. Alonso y Carlos M. Travieso. "Spider Recognition by Biometric Web Analysis". IWINAC 2011, Part II, LNCS 6687, pp. 409-417, 2011. Springer-Verlag Berlin Heidelberg 2011.
- [5] González R., Woods R. : "Digital Image Processing". Library of Congress Cataloging-in-Publication Data, pp. 116-137. Second edition (2002)
- [6] W. Malina, S. Ablameyko , W. Pawlak. "Fundamental Methods of Digital Image Processing", 2002.
- [7] Nobuyuki Otsu (1979). "A threshold selection method from gray-level histograms". IEEE Trans. Sys., Man., pp:62-66.
- [8] J. K. Anil. "Fundamentals of digital image processing", Prentice Hall, 2004.
- [9] Jolliffe, I.T.: "Principal Component Analysis", 2nd edn. Springer Series in Statistics (2002).
- [10] Alan V. Oppenheim, Alan S. Willsky, S. Hamid Nawab. "Signals and Systems", Segunda edición. Editorial: Prentice Hall. pp. 358-367, (1998).
- [11] Cortes, C.; Vapnik, V. (1995). "Support-vector networks". Machine Learning

