



Universidad Nacional de
Educación a Distancia

Universidad Pública de
Navarra

Trabajo de Fin de Master

Máster Universitario En I.A. Avanzada: Fundamentos, Métodos
Y Aplicaciones

IDENTIFICACIÓN DEL CONTENIDO DE NITRÓGENO A PARTIR DE FOTOGRAFÍAS DIGITALES

Autor:
Asier Urío Larrea

Director:
Dr. Miguel Pagola
Tutor:
Dr. Manuel García Luque

20 de febrero de 2012

AGRADECIMIENTOS:

Quiero expresar mi agradecimiento a las siguientes personas:

A mi tutor en la UPNA Miguel Pagola, por su ayuda durante el trabajo.

Al Cesar Arrese-Igor y al grupo de Fisiología Vegetal, por proporcionar los datos del material vegetal.

A la UNED, por darme la oportunidad de hacer este Máster.

A mi madre, a Cecilia y a Irati, por estar siempre ahí.

Muchas gracias a todos.

Índice

1. Introducción	2
2. Objetivos	5
3. Material y métodos	6
3.1. Descripción de los experimentos en invernadero	6
3.2. Medidas estándar de las plantas	8
3.3. Fundamentos del color	9
3.4. Adquisición y procesamiento de las imágenes digitales	14
3.5. Cálculo de índices de verdor	16
3.5.1. Índices obtenidos con el nivel de intensidad medio de toda la imagen	18
3.5.2. Índices obtenidos con el nivel medio de cada zona utilizando el algoritmo Watershed	19
3.5.3. Índices obtenidos mediante la programación genética	23
3.5.4. Redes Neuronales	29
3.5.5. Tratamiento de los datos	31
4. Resultados y discusión	33
4.1. Índices obtenidos con el nivel de intensidad medio de toda la imagen . . .	34
4.2. Índices obtenidos con el nivel medio de cada zona utilizando el algoritmo Watershed	40
4.3. Índices obtenidos mediante la programación genética	43
4.4. Redes Neuronales	48
5. Conclusiones	51

Capítulo 1

Introducción

El estado nutricional de una planta es determinante para obtener una buena producción en cantidad y en calidad. Una mala salud del cultivo provocará un decremento en la calidad y producción del mismo. El estado nutricional de una planta se ve afectado por los nutrientes, el agua que puede tomar del entorno y por las condiciones ambientales como la luz y la temperatura.

Una vez que un cultivo tiene cubiertas las necesidades hídricas, el nitrógeno es el nutriente que más condiciona la producción tanto en entornos naturales como agrícolas [15]. La fertilización nitrogenada es un factor que no puede faltar en ningún cultivo, mientras que el resto de nutrientes tienen un papel más secundario, ya que si no se cubren las necesidades de nitrógeno el resto de nutrientes no hacen efecto.

Agronomicamente es muy importante poder evaluar el estado nutricional de una planta, especialmente de nitrógeno, para poder fertilizar las plantas de un modo óptimo y obtener así una gran producción con un máximo de calidad. El no evaluar correctamente las necesidades de fertilización puede acarrear problemas por exceso o defecto de la misma; una fertilización nitrogenada excesiva hará que el cultivo sea menos rentable por haber gastado más nitrógeno además de una mayor degradación ambiental y una reducción en la calidad del cultivo [17, 21]. Mientras que una fertilización escasa no permitiría obtener la cantidad de cultivo esperada, o incluso sería de peor calidad.

Además hay que considerar que el consumo de vegetales afecta de forma muy importante de la dieta [19]. Por tanto es conveniente un uso racional de la fertilización nitrogenada para disminuir los riesgos de la salud debidos a una excesiva acumulación de nitratos [4]

Por tanto la estimación del contenido de nitrógeno es de vital importancia en el desarrollo de los cultivos, tanto desde el punto de vista agronómico (de producción),

como desde el sanitario.

Según [25], la clorofila extraíble está relacionada positivamente con la concentración de nitrógeno en las hojas. Una alta concentración del contenido de clorofila de una planta se debe a una mayor densidad de ésta en los cloroplastos, más que a una mayor cantidad de éstos [26]. La cantidad de clorofila puede estimarse a su vez por distintos métodos.

Los más exactos son los procesos químicos, que a partir de muestras del material vegetal y tras un proceso de varios pasos en el laboratorio logran obtener de una manera muy precisa la cantidad de clorofila presente en la muestra. Esta determinación se realiza mediante la extracción de la clorofila del material vegetal, gracias a la utilización de disolventes orgánicos y su posterior análisis mediante imágenes. Sin embargo estos son muy caros y lentos para un uso extensivo, además de ser métodos destructivos que anulan la parte de la planta utilizada para realizar el análisis. Estos métodos han sido los más utilizados, sobre todo en trabajos de investigación.

Otro de los métodos se basa en un aparato medidor de clorofilas, el SPAD-502 . Este determina la cantidad relativa de clorofila presente mediante la medición de la absorción de la hoja en dos regiones de longitud de onda; en las regiones roja y cercana a la infrarroja. Utilizando estas dos transmisiones el medidor calcula el valor numérico SPAD que es proporcional a la cantidad de clorofila presente en la hoja y en consecuencia de Nitrógeno (N). Este método ha ido sustituyendo progresivamente a los análisis químicos, sobre todo en trabajos de campo, y más lentamente y en combinación con ellos en la investigación básica.

El uso del SPAD-502 y su relación con el contenido de nitrógeno de la planta ya ha sido documentado para plantas de espinaca [13], y para plantas de guisante [10].

Sin embargo este método, pese a su facilidad, presenta también varios inconvenientes [8]. El más importante de ellos es que pese a observar una gran correlación entre el valor SPAD y el contenido de clorofila de la hoja, la ecuación que relaciona ambos valores es dependiente de la especie vegetal y no es siempre la misma, o incluso el propio estado de salud de la planta también afecta a esta medida. Además el aparato tiene un precio elevado que no permite su uso en explotaciones agrícolas individuales.

El contenido de clorofila de las hojas de las plantas a su vez también está relacionado con el color (verdor) que estas presentan a simple vista; donde un mayor verdor de la hoja indica una mayor concentración de clorofilas en la misma. Por este motivo varios autores han propuesto diferentes índices, que usan de los valores de intensidad R, G y B en los píxeles de la imagen, que se relacionan con la cantidad de nitrógeno.

[11] desarrollaron un índice para la determinación de la cantidad de clorofilas en

plantas de trigo y centeno mediante el uso de una videocámara sobre todo el cultivo. El sistema consiste en grabar el cultivo con una videocámara y una vez digitalizadas las imágenes en el ordenador seleccionar una hoja que se vea correctamente en cada imagen. Posteriormente a partir de los valores de cada imagen se obtiene el índice que relaciona los valores de la imagen con el contenido de clorofilas.

En el citado trabajo realizado por Pagola et. al. [20] se presenta un método basado en el análisis de los componentes principales de las diferencias entre los valores R, G y B de los píxeles de la imagen. En este caso desarrollaron el sistema mediante el uso de una cámara de fotos digital y en plantas de cebada.

Para el mismo tipo de planta, cebada, en [9] comprobaron también la validez de los índices RGB y del espacio de colores Lab*. En este trabajo obtuvieron una muy buena correlación entre el índice R+G y el contenido de clorofila, así como con el valor L* del espacio Lab*, unos resultados que se aproximan a los que se obtiene con el SPAD.

En [23], estimaron mediante una regresión lineal la relación entre los valores de SPAD y clorofila con el color de la hoja en plantas de arroz. En este caso el índice que propusieron es el valor medio del verde de la hoja.

El grupo de [2], busco mediante la fotografía digital la posible determinación del contenido de clorofila en plantas de tomate. Para ello tomaron imágenes de las plantas y obtuvieron el valor medio de cada color R, G y B. Con estos datos calcularon la correlación de cada uno de los valores con el contenido de nitrógeno de la hoja, obteniendo como resultado una gran correlación de los valores rojo y azul, y una muy baja correlación con el valor verde.

También en plantas de tomate, en [28] se encontró una correlación entre los valores RGB ajustados ($r = R/(R + G + B)$, $g = G/(R + G + B)$, y $b = B/(R + G + B)$), siendo el resultado de la transformación (rgb) menos dependiente de la luz. En este trabajo obtuvieron una alta correlación entre el valor rojo medio y el contenido de clorofila como en [2], sin embargo no se dio el mismo resultado con el azul, que presento una mala correlación, ni con el verde que en este caso si presento una buena correlación con el contenido de clorofila.

Los resultados obtenidos en los dos últimos trabajos citados muestran la dificultad de obtener índices similares, no solo en distintas especies como puede verse al comparar todos los trabajos, sino incluso, en la misma especie con distintas condiciones experimentales.

En todos estos trabajos queda claro que los índices obtenidos varían según la especie con la que se esta trabajando.

Capítulo 2

Objetivos

Este trabajo tiene como precedentes diversos estudios que relacionan el contenido de nitrógeno de una planta con su imagen.

El proyecto se realizará en colaboración con el departamento de ciencias del medio natural de la UPNA en donde se disponen los recursos vegetales.

Los objetivos que se pretenden conseguir con la realización de este trabajo son:

1. Buscar un índice que relacione los valores obtenidos a partir de la imagen digital con el contenido de clorofila de la hoja, para los casos de espinacas y guisantes.
2. Comprobar la validez de la programación genética para obtener una expresión matemática utilizando las tres componentes del color RGB de una imagen de una hoja, prediga el contenido en clorofila de la hoja.
3. Comprobar la validez de las redes neuronales para obtener una red que prediga el contenido de clorofila de una hoja utilizando las componentes RGB de la imagen de la hoja.

Capítulo 3

Material y métodos

Para poder buscar una relación entre los valores R,G,B de la imagen de la hoja y el contenido de clorofila de la misma, es necesario obtener un número de muestras suficiente en las que se obtenga un rango de colores y contenidos de clorofila para poder crear así una relación entre los mismos. Cuanto más rango abarquen estos valores más consistente será el índice encontrado, ya que es mejor operar interpolando en el rango conocido, que extrapolando a posibles valores que no han sido medidos. Por tanto es necesario crear experimentalmente una situación en la que las plantas produzcan una amplia gama de colores y contenidos en clorofila.

El experimento en invernadero se llevo a cabo en el departamento de Ciencias del Medio Natural de la Universidad Pública de Navarra.

Las plantas se someten a distintos tratamientos que provocaran que estas tengan distinto contenido en clorofila y por tanto también distinto color de hoja, creando así una gama de valores.

En este apartado de material y métodos se describe inicialmente cuales han sido y en que han consistido los tratamientos que se han aplicado a las plantas. Así como la cantidad de muestras que se han generado. Posteriormente se describen las medidas analíticas que se han determinado en las propias plantas con el fin de evaluar su contenido en clorofilas. Finalmente se describen los métodos relativos al procesamiento de la imagen y obtención de los distintos índices.

3.1. Descripción de los experimentos en invernadero

Material biológico: Para este trabajo se han utilizado dos especies; *Spinacia oleracea* var. Butterflay (Espinaca) y *Pisum sativum* var. Lincoln (Guisante). Se seleccionaron

estas variedades por su buena adaptación a la tierra y por su elevado interés agrícola.

Todos los experimentos se llevaron a cabo en el invernadero de la finca de práctica y de investigación agrícola de la escuela Técnica Superior de Ingenieros Agrónomos de la Universidad Pública de Navarra.

Las plantas se cultivaron en macetas de polietileno de 1 litro de capacidad. El sustrato utilizado fue una mezcla al 50 % de turba fertilizada y perlita.

La primera semana de cultivo se realizó en el fitotrón¹, con riego manual cada 1 o 2 días. La segunda semana se pasaron al invernadero nº 1, con riego automático por aspersión y “fog system²”. Las semanas 3^a y 4^a, se trasladaron las plantas al invernadero nº 2, con riego manual cada 1 o 2 días.

Las plantas se sembraron en tres series, en todos los casos dentro de la misma semana. En el caso de las plantas de guisante se realizaron dos tandas (la segunda de ellas con solo dos series) y para las espinacas se realizó una única tanda. En cada una de las series se sembraron 16 macetas con entre 10 y 12 semillas cada una.



Figura 3.1: Conjunto de macetas

Tras la siembra se dejaron crecer por dos semanas y posteriormente fueron sometidas a los respectivos tratamientos.

Los tratamientos que se realizaron son los siguientes:

¹En este tipo de cámaras, se pueden simular condiciones ambientales variables de temperatura y humedad, y también de radiaciones solares y atmósferas gaseosas modificadas (ozono, CO₂, etc.) en función de los entornos de investigación que se pretendan estudiar.

²Sistema nebulizador de agua que permite mantener la humedad en el invernadero.

- Control: Las plantas se dejaron crecer normalmente con un riego adecuado. Este es el tratamiento de referencia.
- Salinidad: En este caso no se trata de un único tratamiento, si no 3, consistían en regar durante una semana, de forma manual hasta capacidad de campo cada 2 días con una solución de NaCl a diferentes concentraciones, siendo finalmente tres tratamientos distintos; Salinidad moderada 6g/l, salinidad intensa 12 g/l y salinidad severa 18 g/l.
- Sequía: Las plantas de este tratamiento se dejaron sin regar. Este tratamiento también tuvo dos grados; Sequía I, con una duración de 6 a 7 días y Sequía II con una duración de 9 a 10 días.
- Encharcamiento: Para realizar este tratamiento se sellaron los orificios de drenaje de las macetas y también se puso una bandeja para que el encharcamiento fuera más efectivo y el sistema radicular estuviera encharcado totalmente.
- Oscuridad: En este caso después de tres semanas de crecimiento normal, se trasladaron a un armario sellado con el objetivo de privarlas de la radiación fotosintéticamente activa para que la pérdida de pigmentos fuera máxima. El riego se realizó de forma manual cada 2 días.

3.2. Medidas estándar de las plantas

Para cada tratamiento y serie se realizaron las siguientes determinaciones analíticas, tomando una muestra por tratamiento y serie en cada cultivo.

- Peso fresco (PF): Peso de las hojas contrarias a las hojas más jóvenes totalmente expandidas, nada más separarlas de la planta, es el peso que tienen mientras están en la planta con vida.
- Peso turgente (PT): Peso de las mismas hojas utilizadas para la determinación del peso fresco tras introducirla en un tubo de ensayo sellado con 10 ml de agua destilada. Es, por tanto, el máximo peso que pueden conseguir las hojas.
- Peso seco (PS): Es el peso de las mismas hojas tras estar durante 48 horas en una estufa a 80 °C. Con los datos anteriores se determina el contenido hídrico y el contenido hídrico relativo de las hojas:

- $CH \% = \frac{PF-PS}{PF} \cdot 100$
- $CHR \% = \frac{PF-PS}{PT-PF} \cdot 100$
- Minolta SPAD-502: Las mediciones con este aparato se realizaron en primer lugar en las mismas hojas más jóvenes totalmente expandidas, que se utilizarían para la realización de las medidas del área foliar y de la extracción de pigmentos.
- Área foliar (AF): La determinación del área foliar se realizó mediante un medidor de superficies (Li-3000 A) que determina la superficie en cm^2 .
- Área foliar específica (SLA): Se obtiene de la relación entre el área foliar y el peso fresco.

$$SLA = \frac{AF}{PF} \text{ (m}^2\text{/kg)}$$

- Extracción de pigmentos: Las muestras se homogeneizaron en 10 ml de acetona al 80 % en un mortero y el extracto se centrifugó a 5.000 rpm durante 5 minutos en una balanza de sobremesa (Selecta). Posteriormente se filtro el sobrenadante mediante filtros de acetato de celulosa de $0,2 \mu m$. A continuación se midió la absorbancia en un espectrofotómetro a longitudes de onda de 750nm para comprobar si existía materia orgánica en suspensión, 663,2 nm y 646,8 nm como máximos de absorbancia de la clorofila a y b respectivamente y 470 nm para determinar el contenido de carotenoides.

La concentración de clorofilas y carotenoides se calculo según la formulas descritas por Lichtenthaler (1987):

- Clorofila a (C_a) = $12,25 \cdot A_{663,2} - 2,79 \cdot A_{646,8}$
- Clorofila b (C_b) = $21,50 \cdot A_{646,8} - 5,10 \cdot A_{663,2}$
- Clorofila a+b (C_a+b) = $7,15 \cdot A_{663,2} + 18,71 \cdot A_{646,8}$
- Carotenoides (C_x+c) = $\frac{10,000 \cdot A_{470} - 1,82 \cdot C_a - 85,02 \cdot C_b}{198}$

3.3. Fundamentos del color

El fenómeno físico del color se basa en la radiación electromagnética, que se caracteriza por su longitud de onda y su intensidad. Cuando la longitud de onda se encuentra entre 390 nm y 750 nm, el ojo humano puede percibirla y se conoce como luz visible.

Las fuentes de luz emiten, generalmente, luz a diferentes longitudes de onda, en función de su temperatura.

Cuando la luz solar atraviesa un prisma de cristal se descompone en los colores básicos del espectro visible.

El color de los objetos

El color de un objeto depende tanto de sus características físicas en su entorno como del sistema que lo percibe, generalmente el ojo y el cerebro. Se puede decir que los objetos tienen el color de la luz que abandona sus superficies, que depende del espectro de la iluminación incidente y las propiedades reflectoras de sus superficie, así como de los ángulos de iluminación y captación. Sin embargo, algunos objetos, no solo reflejan luz, sino que la transmiten o incluso la emiten ellos mismos, contribuyendo de esta forma al color.

El color que percibe una persona está determinado por la naturaleza de la luz que refleja el mismo. Un objeto que refleja todas las longitudes de onda se muestra blanco, en cambio un objeto que refleje solo un rango limitado de longitudes de onda mostrará un determinado color. Por ejemplo los objetos verdes reflejan la luz con longitudes de onda en el rango de 500 a 570nm y la absorben en otras longitudes de onda.

En el caso de la vista humana la percepción de un mismo color puede verse afectada por el entorno del mismo, pudiendo percibir como diferentes dos colores iguales.

Un ojo humano ve los colores como una combinación de los tres colores primarios rojo (R), verde (G) y azul (B). Estos colores están estandarizados desde 1931 por la CIE (*Comisión Internationale de l'Eclairage*) de la siguiente manera: azul = 435.8 nm, verde = 546.1 nm y rojo = 700 nm. Estos colores primarios pueden mezclarse y conseguirse así los colores secundarios magenta (R+B), cyan (G+B) y amarillo (R+G). Una combinación a partes iguales de los tres colores primarios, o de los tres secundarios produce el color blanco.

Cuando se diferencian los colores se hace en tonro a tres características llamadas brillo, matiz y saturación. El brillo es el equivalente a la intensidad, el matiz está asociado a la longitud de onda dominante y la saturación es la cantidad de luz blanca mezclada con el matiz.

Como resumen, el color de un objeto es resultado de sus propiedades superficiales, sus propiedades de transmisión y sus propiedades de emisión, contribuyendo todas ellas a la mezcla de longitudes de onda que abandonan la superficie del objeto.

Pigmentos

Los pigmentos son compuestos químicos que absorben y reflejan distintos espectros de luz. Cuando una superficie tiene un pigmento, la luz que incide sobre la misma es reflejada excepto algunas longitudes de onda. Esta substracción de longitudes de onda es la que produce los diferentes colores. Los colores primarios de los pigmentos son magenta, cyan y amarillo y los secundarios rojo, verde y azul. En el caso concreto de las plantas, las hojas tienen mayoritariamente un pigmento llamado clorofila, que refleja la luz verde y absorbe el resto de las longitudes de onda, razón por la que se ven de color verde. Si bien las plantas también tienen otros pigmentos llamados carotenoides que reflejan colores más rojizos, pero estos están en menos cantidad y su efecto no es tan notorio.

Sistemas de color

A la hora de captar o generar colores disponemos de aparatos aditivos y sustractivos. Los aditivos unen varios espectros para formar un color como los tubos de rayos catódicos y los sustractivos atenúan porciones del espectro de iluminación como las impresoras de inyección de tinta.

Sistemas de color aditivos

En estos sistemas los colores se generan combinando fuentes de luz primarias con diferentes longitudes de onda. Se suelen usar unos valores de control para las cantidades de intensidad de los primarios y relacionar así la salida con los valores o colores que se tienen como objetivo o meta. (R G + B = imagen de color)

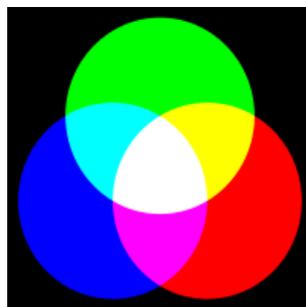


Figura 3.2: *Sistema de color aditivos*

Sistemas de color sustractivos

Se fundamentan en la propiedad de que el color se puede obtener quitando porciones de un espectro fuente. Cada colorante absorbe su color complementario transmiten el espectro restante. En las tintas el espectro que finalmente ve el observador es el producto de una fuente de iluminación, la transmisión de la tinta y la reflectancia del papel. En medios reflectantes sería necesario una matriz diagonal que representaría el espectro de la reflectancia de la superficie. (Luz blanca - Amarillo - Magenta - Cyan = Imagen de color)



Figura 3.3: *Sistema de color sustractivo*

Modelos de color

Con el objetivo de estandarizar la especificación de los colores se crean distintos modelos de color. Matemáticamente se puede representar un color como un vector de coordenadas en un espacio tridimensional, tal y como lo haría el ojo humano, de forma que cada color se representa por un punto en ese espacio tridimensional.

Los modelos de color tienen distintas orientaciones, principalmente al hardware (monitores, impresoras,...) o a aplicaciones de manipulación de color (creación de gráficos, animación,...). Los modelos orientados al hardware más utilizados son el RGB (rojo, verde, azul) para monitores y cámaras principalmente, el CMY (cyan, magenta, amarillo) para impresoras y el YIQ (reflectancia, fase, cuadratura) para televisores. En aplicaciones de procesamiento de imágenes se utilizan RGB, YIQ, HSV (matiz, saturación, valor).

RGB

El modelo RGB es el esquema orientado a hardware comúnmente más utilizado para imágenes digitales. Mantiene la compatibilidad con instrumentos que originan o exponen imágenes basadas en la fisiología de la retina humana. Los colores en RGB se obtienen

como la suma de los tres primarios Azul, Verde y Rojo, correspondiendo a la localización de las respuestas del cono máximas a percepciones o estímulos monocromáticos. El modelo esta basado en el sistema de coordenadas cartesianas. Las imágenes de este modelo se forman por la combinación en diferentes proporciones de los 3 colores primarios RGB.

En este tipo de representación el color negro se representa como $(0,0,0)$, ausencia de todo color, y el blanco todo lo contrario $(1,1,1)$. Los colores primarios corresponden a Rojo $(1,0,0)$, Verde $(0,1,0)$ y Azul $(0,0,1)$; mientras que otros ejemplos de colores serían: Cyan $(0,1,1)$, Magenta $(1,0,1)$ o Amarillo $(1,1,0)$. Y entre ellos todos los rangos posibles. Frecuentemente los valores RGB se representan como enteros entre 0 y 255, en lugar de reales entre 0 y 1.

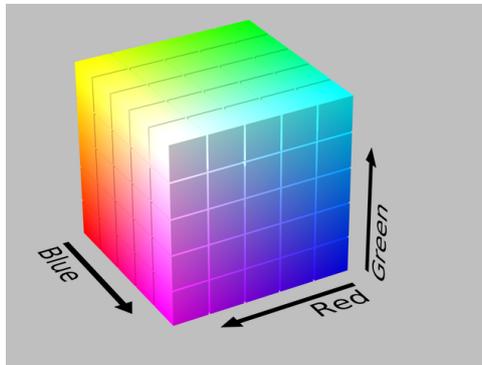


Figura 3.4: *RGB*

CMY

Cyan, magenta y amarillo son los colores secundarios de la luz, o los primarios de los pigmentos. Si se hace incidir luz blanc en un objeto pintado de amarillo, este absorbe la luz azul y no la refleja. Es decir, el amarillo subtrae la luz azul de la blanca.

YIQ

Es el modelo que se utiliza en la televisión. Es una combinación de los valores RGB, que se obtiene de la siguiente manera:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0,299 & 0,587 & 0,114 \\ 0,596 & -0,275 & -0,321 \\ 0,212 & -0,523 & 0,311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Este modelo se diseño teniendo en cuenta que el sistema visual humano es más sensible a cambios en la reflectancia que en el matiz o saturación.

La principal ventaja del modelo es que la reflectancia (cantidad de luz percibida por el ojo) y la información del color se pueden tratar por separado.

HSV

Otra representación habitual es el HSV (Matiz, Saturación y Brillo), que se representa mediante un cono donde el eje del cono representa la línea de gris. El matiz describe la pureza del color y la saturación una medida del grado en el que el color puro es diluido con la luz blanca. El componente de intensidad se puede separar de la información del color. Además los componentes de matiz y saturación están muy relacionados con la forma en que los humanos perciben el color.

Los valores HSV se pueden obtener a partir de los RGB mediante la siguiente transformación:

$$V = \max(R, G, B)$$

$$S = \frac{\max(R, G, B) - \min(R, G, B)}{\max(R, G, B)}$$

$$H = \arccos \frac{\frac{1}{2} \cdot (2R - G - B)}{\sqrt{(R - G)^2 - (R - B)(G - B)}}$$

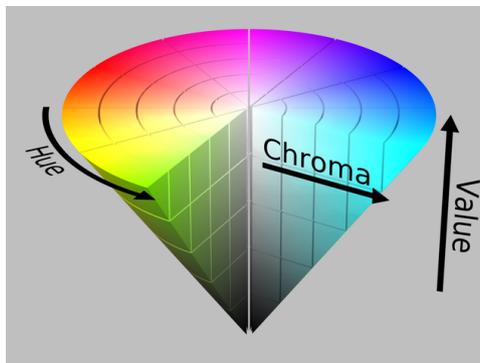


Figura 3.5: Cono HSV - Hue: Matiz, Chroma: Saturación y Value: Intensidad

Los sistemas de representación del color RGB y HSV serán los que se usen en este trabajo. Por ser los más mencionados en la literatura.

3.4. Adquisición y procesamiento de las imágenes digitales

Todo el procesamiento de las imágenes se realizó mediante el programa MATLAB y su librería Image Processing Toolbox.

Adquisición de las imágenes digitales

La adquisición de las imágenes se realizó en las mismas hojas que se utilizaron para tomar las medidas con el SPAD-502, área foliar y extracción de pigmentos. Las fotografías se tomaron al mismo tiempo que las mediciones SPAD.

Las fotos se tomaron con una cámara Pentax Optio A10, la cámara tiene una resolución de 8 megapixels, obtenida mediante un sensor CCD. La captura se realizó poniendo en torno a la parte de la hoja a fotografiar una cartulina negra con un cuadro libre de 1 cm². El objetivo se situó a 20 cm de la hoja bajo la luz artificial del laboratorio, sin utilizar flash.

Filtrado de la imagen.

Como se ha descrito anteriormente cada imagen tiene un cuadro de 1cm² con la hoja y esta rodeada de una cartulina negra. El primer paso es por lo tanto eliminar de la imagen la cartulina para trabajar únicamente con la hoja.



Figura 3.6: *Imagen original*

Para esto se creó un filtro que eliminaba la parte carnosa de la imagen y resaltaba el verde de la hoja:

- $|R - G| - |G - B| < 5$ Este primer filtro tiene por objetivo seleccionar las regiones con más verde de la fotografía. (en la imagen 3.7, se ha sobreseleccionado debido al brillo de la cartulina).
- $|G - B| > 40$ Este segundo filtro tiene como función el eliminar el color carne de la imagen, y también elimina todos aquellos colores que están fuera del rango típico de la hoja. (Imagen 3.8).

Los valores 5 y 40 de ambos filtros han sido obtenidos experimentalmente.

El resultado de aplicar los filtros es una imagen binaria por cada uno de ellos, en los que el valor 1 indica que se cumple el criterio establecido y el valor 0 que no se cumple (en las imágenes de la página 17, cero es negro y uno es blanco).

Una vez pasada la imagen por ambos filtros, cuyo resultado es una imagen binaria, los resultados se multiplican para obtener una imagen que combine las bondades de ambos (Imagen 3.9), obteniendo una imagen en la que prácticamente solo aparece el recuadro de la hoja. Aún así, todavía puede haber imágenes en las que haya pequeñas zonas dentro del recuadro que estén negras, es decir, que no se consideren como parte de la hoja, o bien que suceda lo contrario en zonas fuera del recuadro.

Para solventar esas situaciones se procede a erosionar y dilatar la imagen haciendo que los píxeles que están más aislados se fundan con su entorno. Con el mismo fin se rellenas los huecos menores de 25000 píxeles, tamaño seleccionado experimentalmente, que es siempre menor que el recuadro de la hoja. En esta fase también tiene como objetivo limpiar los bordes de la imagen, zona que también ha podido quedar marcada como parte a considerar. El resultado puede verse en la imagen 3.10.

Una vez que la imagen se ha limpiado y solo queda el cuadro con la hoja se calcula la inclinación que tiene el rectángulo que lo circunscribe y según este ángulo se rota la imagen y finalmente se recorta la misma tomando como base el rectángulo circunscrito eliminando un 25 % a partir de los márgenes, debido a que tras las modificaciones que se le han aplicado a la imagen el rectángulo circunscrito abarca también parte de la cartulina, y como efecto secundario positivo se eliminan los bordes que pueden tener algo de sombra y alterar el color.

Todos estos cálculos se realizan sobre la imagen secundaria binaria y a partir de los parámetros obtenidos de esta (posición e inclinación del rectángulo) se procede al recorte en la imagen RGB original.

Todo el código esta correspondiente al filtrado de la imagen en el archivo recortar.m, página 54.

3.5. Cálculo de índices de verdor

Las imágenes digitales pueden tener distintos modelos de representación en el ordenador. Uno de ellos, que es de los más utilizados tanto en la práctica como en la literatura citada, es el modelo RGB (Rojo, Verde, Azul), que se ha explicado anteriormente.



Figura 3.7: $|R-G|/|G-B|$

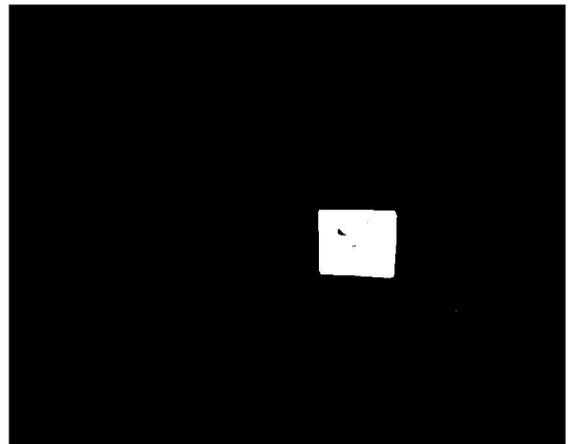


Figura 3.8: $|G-B|$

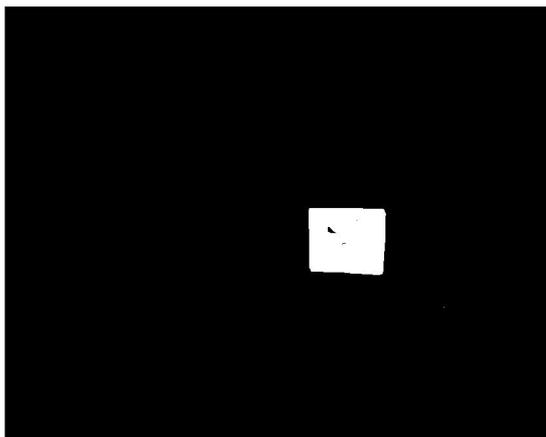


Figura 3.9: *Combinación de los filtros*

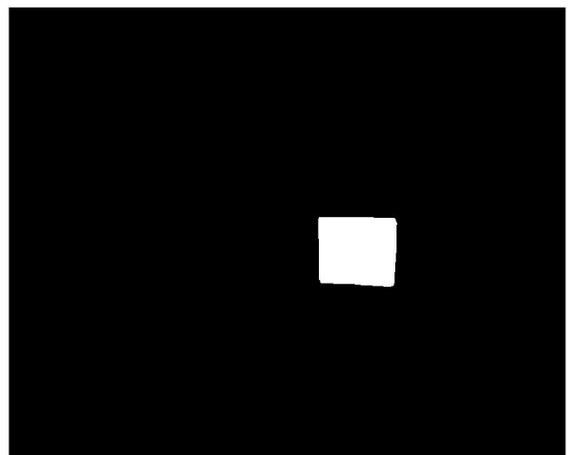


Figura 3.10: *Perfeccionamiento*



Figura 3.11: *Recorte final*

3.5.1. Índices obtenidos con el nivel de intensidad medio de toda la imagen

Primeramente se calculan directamente a partir de las imágenes filtradas, una serie de índices basándose en el valor medio de los píxeles de la imagen. Los índices que se calculan se basan en otras propuestas de la literatura, fundamentalmente en [2], y sus variantes en el espacio de representación HSV.

A partir de cada imagen filtrada se obtuvieron estos valores medios, tanto para el modelo de representación del color RGB como para el HSV.

Una vez calculados los valores medios de cada imagen se consideraron los siguientes índices:

- | | | |
|--|-------------------------|-------------------|
| ▪ R | ▪ G - B | ▪ V |
| ▪ G | ▪ $(R-G)/(R+G)$ | ▪ $H/(H+S+V)$ |
| ▪ B | ▪ $(R-B)/(R+B)$ | ▪ $S/(H+S+V)$ |
| ▪ $0.7582* R-B -$
$0.1168* R-G +0.6414* G-$
$B $ | ▪ $(G-B)/(G+B)$ | ▪ $V/(H+S+V)$ |
| ▪ $(R-B)/(R+B)$ | ▪ $(R-G)/(R+G+B)$ | ▪ H-S |
| ▪ $R / (R+G+B)$ | ▪ $(R-B)/(R+G+B)$ | ▪ H-V |
| ▪ $G / (R+G+B)$ | ▪ $(G-B)/(R+G+B)$ | ▪ S-V |
| ▪ $B / (R+G+B)$ | ▪ $(R-G)-(G-B)$ | ▪ $(H-S)/(H+S)$ |
| ▪ R/G | ▪ $(R-G)-(G-B))*(G-B)$ | ▪ $(H-V)/(H+V)$ |
| ▪ G/R | ▪ $((R-G)-(G-B))/(G-B)$ | ▪ $(S-V)/(S+V)$ |
| ▪ R - G | ▪ $ G-R +2* R-B $ | ▪ $(H-S)/(H+S+V)$ |
| ▪ R - B | ▪ H | ▪ $(H-V)/(H+S+V)$ |
| | ▪ S | ▪ $(S-V)/(H+S+V)$ |

Estos índices son los utilizados o mencionados en otros trabajos, fundamentalmente en [11], y otros de elaboración propia.

3.5.2. Índices obtenidos con el nivel medio de cada zona utilizando el algoritmo Watershed

Otro de los aspectos a estudiar, que ya lo proponen los autores en el artículo [20], es la mejora en la extracción de las regiones de interés de la hoja, eliminar nervios y zonas que no presentan clorofila de manera natural y que por estar en mayor o menor proporción afectan al cálculo del índice.



Figura 3.12: *Imagen inicial*

Sharafenko [24] propone la utilización del algoritmo watershed para la segmentación de imágenes graníticas en color, imitando la percepción humana de los bordes entre colores. Además este algoritmo crea distintas regiones de la imagen que se utilizan, en este trabajo, para la obtención de valores más significativos, eliminando la influencia de los nervios de la hoja.

El algoritmo Watershed pretende buscar los bordes de las regiones a partir de unos puntos, denominados semilla, sobre los que se va agrandando la región en su entorno hasta que dos regiones se tocas o se solapan, en ese momento se determina el borde entre ellas. Por tanto el resultado del método Watershed es el de una imagen con regiones en torno a las semillas iniciales. El algoritmo devuelve una imagen con valores que indican la pertenencia a una u otra región.

Para ello se desarrolla un algoritmo de segmentación watershed con ciertas modificaciones, ya que el algoritmo que incluye Matlab crea una sobresegmentación (imagen 3.15); para comenzar se trata de obtener sólo las más 20-30 semillas más significativas, para ello

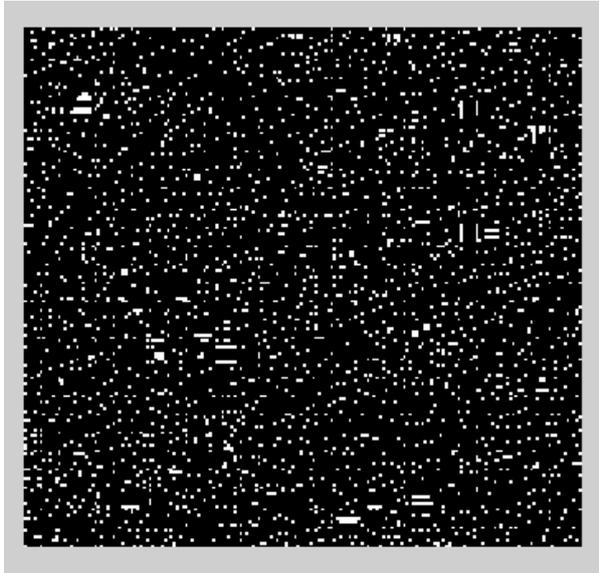


Figura 3.13: *Todas las semillas*

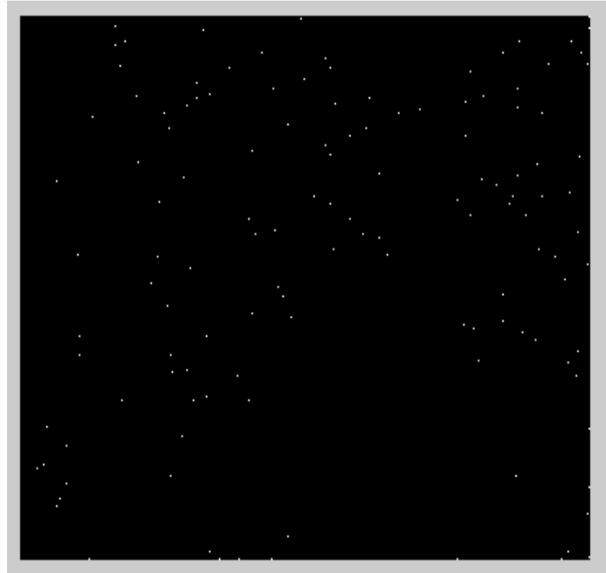


Figura 3.14: *Semillas seleccionadas*

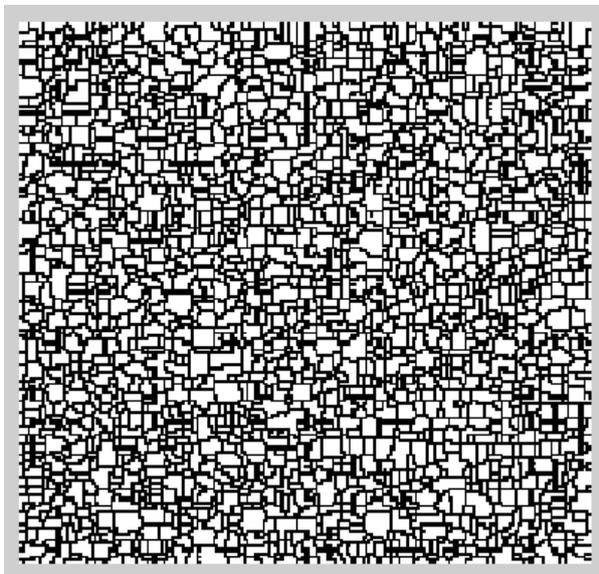


Figura 3.15: *Regiones watershed Matlab*



Figura 3.16: *Regiones watershed finales*

se dilata y erosiona la imagen hasta obtener las semillas finales y así conseguimos de 2.000 a unas 20-30 semillas (se pretenden dejar 20-30 regiones), finalmente se llenan estas regiones con el tono de verde de la foto original correspondientes a las semillas (imagen 3.16), con lo que conseguimos 2 objetivos:

- Eliminamos los tonos de los bordes (de nervios y efectos luminosos)
- Tomamos los tonos más significativos, ya que las semillas son los mínimos del gradiente de la imagen y por tanto pertenecen a las zonas con menor variación de color o tonos de la hoja, es decir a las zonas más alejadas de nervios y bordes, donde la clorofila es más significativa.

Pseudocódigo 1 Búsqueda de los mínimos de la imagen

Entrada: A = imagen RGB.

Salida: Vector con las coordenadas de los mínimos y el valor de verde de los mismos correspondiente a la imagen original.

```

1: invertir imagen
2:  $I \leftarrow$  restar imagen invertida a la imagen original
3:  $vecino \leftarrow 1$ 
4: mientras  $vecinos \neq 0$  hacer
5:    $vecino \leftarrow 0$ 
6:    $esqueletizar(I)$ 
7:   para todo píxel (i,j) de la imagen si  $I(i, j) \neq 0$  hacer
8:     para cada 8-vecino del píxel si  $I(vecino) \neq 0$  hacer
9:        $vecino \leftarrow vecino + 1$ 
10:    fin para
11:   fin para
12: fin mientras
13: para todo píxel (i,j) de la imagen hacer
14:   si  $I(i, j) \neq 0$  entonces
15:      $semillas(h) \leftarrow k, i, j, A(i, j, 2)$ 
16:      $h \leftarrow h + 1$ 
17:   fin si
18:   Nueva columna  $k \leftarrow k + 1$ 
19: fin para
20: devolver semillas

```

Pseudocódigo 2 Inundación de las regiones watersheed

Entrada: A = imagen RGB.**Salida:** Imagen separada por las regiones watersheed, con el color de cada región igual al de la semilla en la imagen A original

```
1: fin ← 0
2: semillas ← minimos(A)
3: buffer ← semillas
4: mientras fin=0 hacer
5:   i ← 1
6:   mientras buffer(i)=0 and i < 256 hacer
7:     i ← i + 1
8:     4 - vecinos ← coordenadas - buffer(i)
9:     para j = 1 to 4 hacer
10:      si no clasificado entonces
11:        si buffer sin elementos entonces
12:          primer elemento
13:        si no
14:          encolar
15:        fin si
16:      fin si
17:    fin para
18:  fin mientras
19: fin mientras
20: para todo pixels hacer
21:   colorA2 = colorA-semilla-ID
22: fin para
23: devolver A2
```

3.5.3. Índices obtenidos mediante la programación genética

La computación evolutiva es un conjunto de técnicas que basadas en ideas obtenidas de la biología, principalmente de la teoría de la evolución, tienen como objetivo el optimizar funciones matemáticas.

El conjunto de algoritmos evolutivos no es determinista como los habituales, si no que se basa en gran medida en el azar. El azar es quien determina en mayor o menor medida que individuos de la población se emparejan, con que frecuencia y en que punto. Esta idea del cruzamiento, así como la mutación, tiene inspiraciones en biología. De hecho, durante la ejecución de un algoritmo evolutivo hay un conjunto de soluciones (más o menos validas, según el tipo de problema), llamado población y ese conjunto va evolucionando, mediante la interacción de los llamados individuos (cada solución candidata).

En la imagen 3.17 se puede observar el esquema típico de un algoritmo evolutivo. A partir de una población inicial, generalmente aleatoria, se seleccionan unos individuos y se cruzan entre si. Posteriormente a la descendencia obtenida se le aplica una mutación y después de evaluar el resultado se seleccionan los individuos que seguirán en el sistema.

El criterio que se utiliza para la mejora de los individuos es el fitness, que es la función que hay que optimizar. El fitness influye, en la mayoría de las variantes, en la selección de los individuos que van a compartir información y en la selección de los individuos que se mantienen en una población dada.

Las distintas familias de algoritmos evolutivos se diferencian en el tipo de representación que usan para definir el problema, el tipo de cruzamiento y mutación que utilizan, o los criterios de selección.

La programación genética es una técnica introducida por J. R. Koza [12], en la que mediante el uso de arboles de expresiones-S, y con ideas basadas en los algoritmos evolutivos (es decir, mediante mutaciones, cruzamientos y selecciones de individuos) se evolucionan formulas aritméticas, lógicas o incluso programas de ordenador.

Las expresiones-S se representan como árboles, en los que las hojas son los llamados elementos terminales (constantes o variables de la aplicación), y el resto son operadores. La mutación y cruzamientos actúan sobre estas expresiones, modificándolas unilateralmente en el primer caso, o bien intercambiando subexpresiones entre dos individuos en el segundo.

En este trabajo el sistema de programación genética que se utiliza tiene como objetivo encontrar una relación matemática que exprese el valor del contenido de clorofila a partir de los valores RGB de la imagen de una hoja. Para ello utiliza el conjunto de terminales R,G y B, los valores del espacio de color y como funciones a utilizar para expresar la

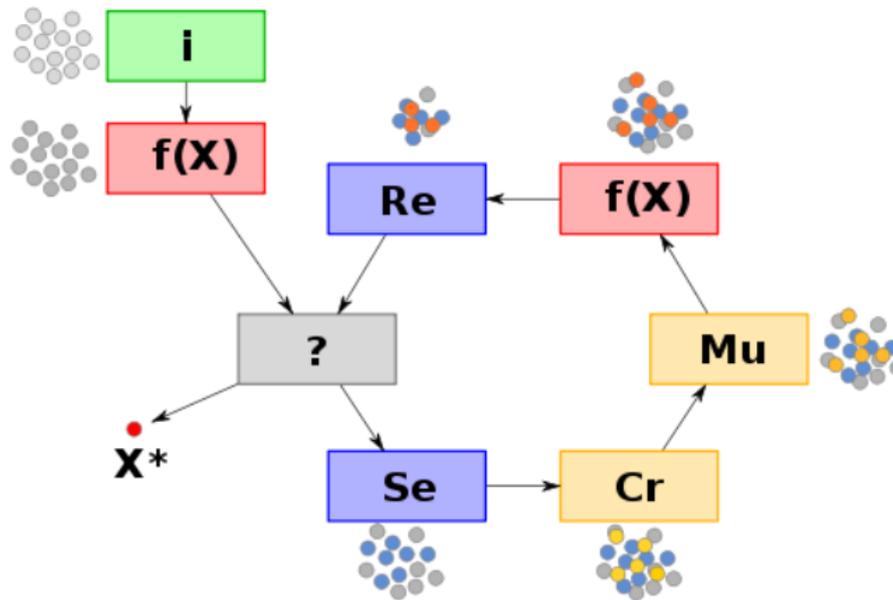


Figura 3.17: Esquema de un algoritmo genético: inicialización, $f(X)$: evaluación, $?$: condición de término, Se : selección, Cr : cruzamiento, Mu : mutación, Re : reemplazo, X^* : mejor solución.

relación entre los valores RGB y el contenido de clorofila se utilizan las siguientes +, -, *, /*, ABS. (donde /* Devuelve 0, al dividir por 0).

Todo ello a través de la librería `gplab.toolbox` [1] para Matlab.

La librería `gplab` es altamente configurable, pudiendo modificarse los siguientes parámetros del algoritmo:

- Tamaño de la población.
- Número máximo de generaciones.
- Función de inicialización.
- **Fullinit.** El árbol va añadiendo nodos no-terminales hasta que se alcanza la profundidad máxima definida, momento en el que se añaden nodos terminales. De esta forma se consiguen unos árboles equilibrados. Cuando se usa el tamaño (número de nodos) en lugar de la profundidad, se añaden no terminales hasta que el tamaño es cercano al máximo y luego solo se añaden terminales. De esta manera no es posible conseguir todos los árboles con el tamaño exacto, solo aproximado, si bien nunca por exceso.

- **GrowInit.** Cada nodo se selecciona aleatoriamente entre no-terminales y terminales, excepto cuando se esta en la profundidad máxima que solo se seleccionan terminales. De esta manera los árboles que se generan pueden estar muy desequilibrados. Si se hace por tamaño, entonces también se seleccionan aleatoriamente, pero cuando se seleccionan cuidadosamente los nodos internos según su aridad para garantizar que no se excede el número máximo.
 - **RampedInit.** En cada nivel se selecciona la mitad de los nodos según el método fullinit y la otra mitad según el growinit. Así se obtiene una población muy diversa.
- **Tamaño de los individuos.** El tamaño de los individuos puede seleccionarse de dos maneras, por la profundidad del árbol o por el número de nodos del mismo. El objetivo del control del tamaño es minimizar en lo posible el efecto bloat, es decir, al aumento indiscriminado del tamaño de la solución sin aportar mejoras a la misma. La forma general de evitar este fenómeno es especificando un tamaño máximo de los nuevos árboles, así al sobrepasarlo en lugar de pasar a la descendencia el nuevo lo haría uno de sus padres.
- **Tamaño inicial.** Es el valor que determina el tamaño inicial de los individuos de la población.
 - **Tamaño máximo estricto.** Cualquier individuo que supere este valor será rechazado y no se incluirá en la próxima generación. En su lugar se incluirá a uno de sus padres.
 - **Tamaño máximo dinámico.** Este valor se tiene en cuenta cuando se permite un tamaño de individuo variable. Es un valor menor o igual que el tamaño máximo estricto y se usa también para limitar el efecto engorde. Si un nuevo individuo es mayor que lo indicado por el tamaño máximo dinámico y menor que el indicado por el tamaño máximo real, se acepta el individuo y se incrementa el valor del tamaño máximo dinámico. Es posible también configurar el programa para que el tamaño máximo dinámico disminuya, limitándolo al tamaño inicial o incluso por debajo de este, siempre que el individuo lo permita.

Con el objetivo de no tener funciones muy largas y que puedan tener significado biológico se limita el tamaño a 20 nodos o a una profundidad de 5 niveles.

- **Funciones y terminales.** Las funciones son los operadores a utilizar sobre los terminales, que son números o variables (que se sustituirán finalmente por números

en el momento de la evaluación). `gplab` puede utilizar cualquier función definida en MATLAB y permite a su vez definir funciones propias. Respecto a los terminales, permite el uso de constantes, números aleatorios en un cierto rango y las variables del problema.

- operadores genéticos. Estos operadores son los utilizados para crear un nuevo individuo a partir de progenitores. La librería implementa los siguientes operadores:
 - Cruzamiento. Se seleccionan nodos aleatorios de cada uno de los dos árboles y se intercambian entre ellos creando así dos descendientes.
 - Mutación. Se sustituye un nodo seleccionado aleatoriamente del árbol por un nuevo árbol generado aleatoriamente mediante el método de crecimiento y se mantienen las restricciones de tamaño de la generación inicial.
 - Mutación encogedora (*shrink*). Se sustituye un subárbol (*S*), por un subárbol de *S*. Excepto en el caso de que *S* sea un terminal, el árbol original decrece.
 - Mutación de intercambio. Se intercambia dos subárboles seleccionados aleatoriamente del mismo individuo.

La librería permite, además, añadir operadores genéticos propios. Cualquiera de los citados operadores no tiene porque devolver descendencia conforme a los criterios de tamaño, ya que esto se evalúa a posteriori.

- Validación de nuevos individuos. Cada vez que se genera un nuevo individuo, hay que evaluar las restricciones de tamaño para ver si es apto para la nueva población. Las funciones que incluye la librería son las siguientes:
 - `strictdepth` y `strictnodes`. Estas rechazan, respectivamente, los individuos con mayor profundidad y mayor tamaño que es estipulado como máximo estricto.
 - `dyndepth` y `dynnodes`. En el caso de que la profundidad o el número de nodos sea mayor que el máximo dinámico, comprueban si el fitness es mejor que el mejor actual, en cuyo caso aceptan el individuo y aumenta el valor máximo dinámico. Si no lo rechazan.
 - `heavydyndepth` y `heavydynnodes`. Son similares a los del punto anterior, pero permiten decrecer el límite dinámico en el caso de que el nuevo individuo tenga un mejor fitness que el mejor encontrado hasta el momento.

Estos filtros pueden rechazar, aceptar o no afectar a los individuos. Cualquier individuo que después de pasar por estos filtros no haya sido aceptado o rechazado, será aceptado como candidato a la nueva población.

- Antes de aplicar un operador genético, como el cruzamiento, hay que seleccionar cuales serán los padres de entre toda la población. Es posible seleccionar cualquiera de los siguientes métodos:
 - ruleta. Se selecciona aleatoriamente, pero cada individuo tiene una probabilidad correspondiente al número de descendencia que se le estima.
 - sus. Como el anterior pero siendo todos los individuos equiprobables.
 - torneo. A partir de una selección aleatoria de un número a determinar de individuos se selecciona el de mayor fitness.
 - lexicotour. Como el anterior, pero en el caso de empate, se selecciona el de menor tamaño con el objetivo de reducir el efecto engorde.
 - doble torneo. Aplica dos capas de torneo. La primera de ellas por fitness y la segunda por tamaño, o al revés.

Es posible añadir funciones propias al repertorio del programa.

- Número de descendencia estimada. Es el cálculo que se realiza para asignar una probabilidad de selección para el método de la ruleta (o similares). Los métodos implementados son los siguientes:
 - absoluto. Proporcional al valor del fitness.
 - rank85. Proporcional a la posición en la población ordenada por fitness.
 - rank89. Como el anterior, pero también influye el estado del algoritmo (la distancia que le separa de la máxima generación permitida). La diferencia entre individuos aumenta conforme se suceden las generaciones.

Como en el resto de parámetros es posible definir funciones de ranking propias.

- Selección de supervivientes. Una vez han sido generados los nuevos individuos, y dado el tamaño finito de la población, hay que seleccionar cuales serán los individuos que formarán la nueva generación. Para ello se ordenan todos los individuos, padres e hijos en función del parámetro de supervivencia elegido, y después los individuos ya ordenados serán seleccionados en función del número de individuos permitidos en la población. La ordenación se puede hacer según los siguientes niveles:

- Reemplazo. Los hijos deben reemplazar a la población anterior, independientemente de que tengan peor fitness. Por esto, se ordenan los hijos en función de su fitness y a continuación los padres, también en función de su fitness. Quedan de esta forma todos los hijos antes que los padres.
 - Mantener el mejor. Se mantienen el mejor padre y el mejor hijo, que son los primeros en la lista, el resto se ordena según el método anterior de reemplazo.
 - medio-elitismo. Se trata de mantener la mejor mitad de cada parte, de padres e hijos. Por tanto son estos los que reciben la mayor probabilidad, el resto se ordenan según el método de reemplazo.
 - elitismo-total. Se ordenan todos en función de su fitness, independientemente de su origen.
- Función de evaluación a utilizar. La librería incluye las siguientes funciones de evaluación del fitness:
- regfitness. Esta función calcula la suma del valor absoluto de la diferencia entre el valor esperado y el obtenido mediante el individuo. Por tanto los mejores individuos son los que presentan valores más bajos, es decir, los que más aproximan su valor al esperado. $\sum |x_i - y_i|$
 - antfitness. Esta función se utiliza para los problemas de simulación de hormigas.
 - antfitnesslib. Otro método alternativo para la evaluación de las hormigas.

Además permite la definición de funciones de evaluación propias.

- Probabilidades de los operadores. La librería incorpora un mecanismo de autoadaptación, que permite que las probabilidades de aplicar los operadores genéticos varíen a lo largo de la ejecución. O se permite mantener unas probabilidades fijas desde el inicio del programa.
- Condiciones de parada. El programa se ejecuta hasta llegar a la máxima generación permitida o hasta que se alcance una condición de parada. en gplab las condiciones de parada se definen de la siguiente manera: [f d], donde f es el porcentaje de individuos que tienen que cumplir la condición d. Permite indicar más de una condición [f1 d1; f2 d2] o ninguna [], siendo en este último caso la única condición el número de generación.

valor	r^2 -clorofilaarea	r^2 -clorofilaarea bootstrap	corrfitness	regfitness
R	0,766	0.768	-0,766	144,535
G	0,290	0.302	-0,3039	141,814
B	0,702	0.687	-0,7054	149,390

Tabla 3.1: *Distintas funciones de fitness y correlación*

Función de evaluación propia

Con el fin de lograr una mejor aproximación de la relación entre los valores RGB de la imagen y el contenido en clorofila de la hoja, he implementado una función de evaluación que en lugar de basarse en la regresión lineal entre las entradas y las salidas, se basa en la correlación que existe entre los valores calculados por la función de cada individuo y los valores esperados de salida.

Tal como se puede ver en la tabla 3.1, los distintos algoritmos seleccionan distintas funciones para aproximar la entrada a la salida. En el caso más simple en el que solo se considera un elemento (por tanto R, G ó B), se puede ver la discrepancia entre ambas funciones. La función regfitness produce un menor fitness en el caso del valor verde (G), ya que la diferencia absoluta entre su valor y el de clorofila/Área es menor, pero en cambio, mediante una regresión es más fiable la aproximación que realiza el rojo (R), que es el elemento que selecciona la función corrfitness.

Cualquier combinación de parámetros no se ajusta igual a distintos tipos de problemas, por lo tanto el primer paso será buscar que combinaciones de estos parámetros producen mejores resultados para los datos disponibles. Para cada combinación estudiada se harán diez repeticiones, calculando la media de los mejores fitness fitness de cada una de ellas. Y almacenando también los mejores resultados, que puedan igualar o superar a los índices propuestos con anterioridad.

3.5.4. Redes Neuronales

Igual que en el caso de los algoritmos genéticos la biología ha inspirado otros aspectos de la inteligencia artificial, como es el caso de la neurología y la computación neuronal. Las redes neuronales se componen en general de tres capas de elementos que a partir de las entradas a la red cada una de ellas realiza una función analógica y pasa el resultado a la siguiente hasta que al final se obtiene un resultado. Este paso de información es

equiparable al de las sinapsis de las neuronas naturales. No hay que olvidar que este modelo proporciona un marco de computación paralela, distribuida, de grano pequeño y autoprogramable.

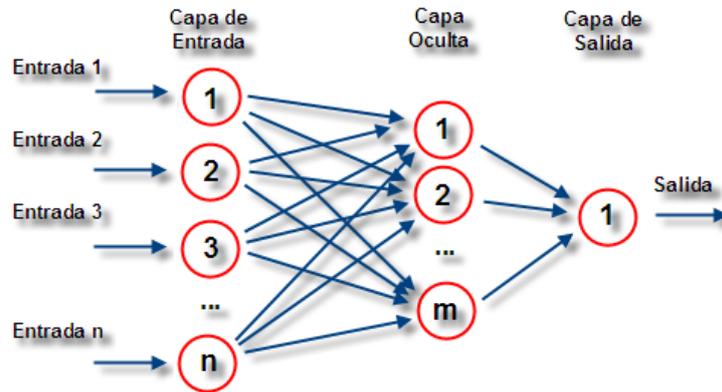


Figura 3.18: Red neuronal artificial perceptrón simple con n neuronas de entrada, m neuronas en su capa oculta y una neurona de escape

Las redes neuronales también se pueden utilizar para la estimación de índices a partir de unas variables independientes de entrada y unas variables dependientes de salida.

Para ello se utilizó la librería de redes neuronales de matlab feedforward. En estas redes de propagación hacia delante o acíclicas todas las señales van desde la capa de entrada hacia la salida sin existir ciclos, ni conexiones entre neuronas de la misma capa. La primera capa tiene una conexión con la entrada de la red, cada capa subsiguiente tiene una conexión con la capa previa y la capa final es la que produce la salida de la red.

A la hora de entrenar la red las entradas se dividen en tres grupos:

- **Entrenamiento:** Esos son los que se utilizan para el entrenamiento propiamente dicho. La red se ajusta en función del error con respecto a estos.
- **Validación:** Se utilizan para evaluar la generalización de la red y para parar el entrenamiento cuando esta generalización no mejora.
- **Comprobación:** Estos no se tienen en cuenta en el entrenamiento y se utilizan para comprobar la validez de la red entrenada.

La **división de los datos** de entrada en estos conjuntos puede hacerse de diversas formas, pero en general se hace de manera aleatoria.

El **entrenamiento** de la red puede utilizar distintas funciones, en este caso se utilizó la función de Levenberg-Marquardt.

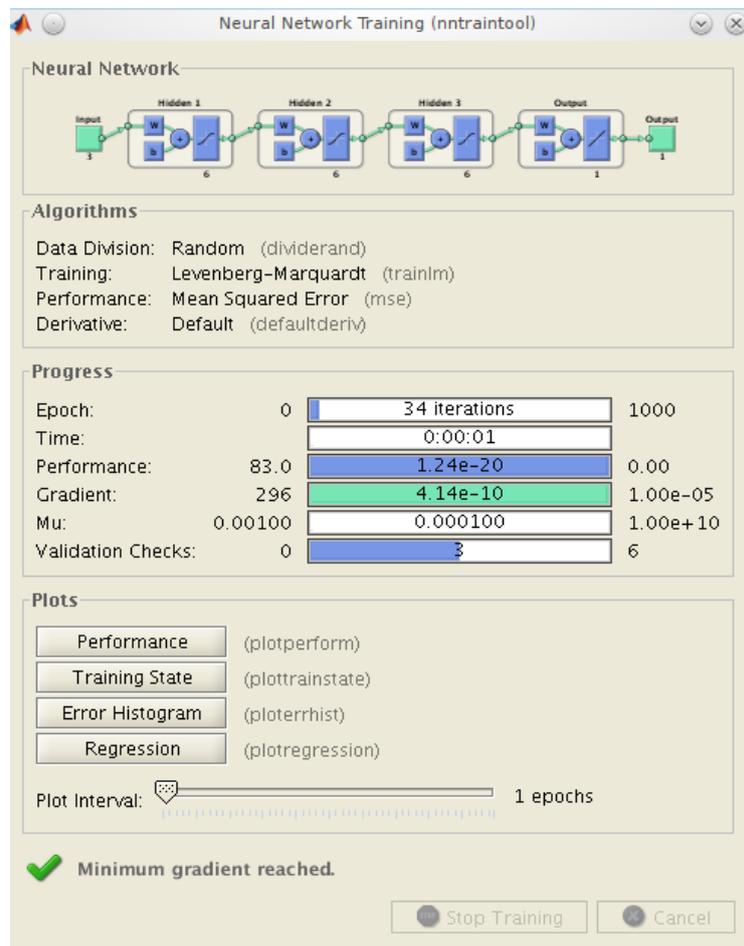


Figura 3.19: Herramienta de trabajo con redes neuronales de MATLAB.

El **ajuste de la red** se puede medir de diversas formas:

- Media de los errores absolutos - Mean absolute error(mae)
- Media del cuadrado de los errores - Mean squared normalized error (mse)
- Suma del cuadrado de los errores - Sum squared error(sse)

Para el ajuste se utilizó la medida del cuadrado de los errores.

3.5.5. Tratamiento de los datos

Para poder comparar los resultados con los realizados por otros autores se utilizará el estadístico r^2 , coeficiente de determinación entre las variables estudiadas. En este trabajo las variables serán el contenido de clorofila y el valor asociado del índice.

Para hacer más robusto el resultado se utilizará la técnica del bootstrapping [6] que consiste en obtener un gran número de submuestras dentro de la propia muestra y poder así aproximar el valor de estadístico. Para ello se utilizará la herramienta proporcionada por MATLAB de la siguiente manera:

```
bootstat = bootstrp(numerodedivisiones,estadístico,variable1,variable2);  
ci = bootci(numerodedivisiones,estadístico,variable1,variable2);
```

Donde el estadístico será el valor r^2 entre las dos variables y las variables serán los valores obtenidos con cada índice y su contenido de clorofila por unidad de área respectivamente.

1. División de la muestra en submuestras y cálculo del estadístico. `bootstat = bootstrp(1000,@rcuad,x,y)`; De esta manera se generan 1000 subdivisiones y se calcula la correlación para cada una de ellas, almacenándose los resultados en la variable `bootstat`.
2. Cálculo de la media del estadístico en las 1000 submuestras. `mean(bootstat)`. De esta forma se obtiene la media de los 1000 estadísticos calculados previamente.
3. Cálculo de la desviación estandar del estadístico en las 1000 submuestras. `std(bootstat)`. De esta forma se obtiene la desviación típica de los 1000 estadísticos calculados previamente.
4. Cálculo del intervalo de confianza al 95 % del estadístico en las 1000 submuestras. `bootci(1000,@rcuad,x,y)`.

`@rcuad` es la función que calcula el r^2 a partir de dos vectores de entrada.

Capítulo 4

Resultados y discusión

Es importante recordar que cada especie tiene unos valores de verdor distintos y por tanto los índices generados no son igualmente validos para cada especie.

Para cada una de las plantas se obtuvieron las medidas de la siguientes forma, en el caso del SPAD se realizaron dos medidas en cada una de las dos hojas y se hallo su promedio; en el caso determinación de la clorofila también se procedió de una manera análoga, promediando el resultado analítico obtenido en cada una de las dos hojas. Por cada planta se hicieron dos imágenes, una por cada hoja, calculando los índices para cada una de las imágenes y obteniendo el promedio de ambas.

Para obtener la correlación de un índice, se analizaban las plantas de cada especie y tratamiento obteniendo valores de 14 macetas, dos por cada una y luego haciendo el promedio como se ha indicado arriba.

En el experimento se contaba con 128 macetas, pero las temperaturas a lo largo de los tratamientos fueron muy variables y los cultivos sufrieron condiciones de crecimiento muy diferentes que afectaron aún más que los tratamientos a su estado. Es por esto que los resultados se podrían haber alterado de manera significativa. Por esto muchas se murieron y otras fueron mal. En otros trabajos se ha visto como condiciones extremas alteran el metabolismo de la planta impidiendo obtener resultados concluyentes.

De todas las plantas utilizadas parece que las espinacas sometidas a oscuridad soportaron mejor las condiciones variables y presentaron unos valores analíticos más acordes con la literatura consultada.

Este apartado se divide en 3 subapartados de las siguiente manera: Primero se estudian los índices propuestos en la literatura sin modificaciones, luego tras la aplicación del algoritmo Watershed, los obtenidos mediante la programación genética y finalmente el uso de las redes neuronales.

4.1. Índices obtenidos con el nivel de intensidad medio de toda la imagen

El coeficiente correlación entre el SPAD y la cantidad de clorofila obtenida para cada especie es de 0,32 en el caso de las espinacas y de 0,3 en el caso de los guisantes.

Tto\Especie	guisante	espinaca
control	0,28	0,3
encharcamiento	0,01	0,07
oscuridad	0,5	0,9
salinidad	0,3	0
sequía	0,5	0,18

Tabla 4.1: *Correlación SPAD - Clorofila ($\mu\text{g}/\text{cm}^2$) por tratamiento y especie*

Vistos los malos resultados obtenidos para cada especie a partir de todas las plantas, se comprueba la correlación existente entre la cantidad de clorofila existente por unidad de área y el valor SPAD correspondiente. Como se puede ver en la tabla 4.1, solo el tratamiento en oscuridad de las espinacas obtiene un valor semejante a los descritos en la literatura [8]. Todo el resto de valores muestran que no existe la correlación que debería entre el valor del SPAD y la cantidad de clorofila presente.

En el gráfico 4.1 se puede ver el diagrama de dispersión para el caso tratamiento de plantas de espinaca y oscuridad que da el mejor y único resultado admisible en cuanto a la relación entre el SPAD y la cantidad de clorofila presente en la hoja.

En la tabla 4.2, puede verse una de las dos imágenes obtenidas para cada planta de espinaca, en el tratamiento de oscuridad y sus valores analíticos asociados. A simple vista destaca como mayores valores analíticos corresponden a imágenes más verdes, mientras que conforme estos valores van disminuyendo las imágenes se hacen más amarillentas.

	SPAD	Clorofila ($\mu\text{g}/\text{cm}^2$)			SPAD	Clorofila ($\mu\text{g}/\text{cm}^2$)
	22,8	18,71			23,4	12,94
	24	13,22			19,4	11,75
	24,7	14,13			14,4	11,6
	24,6	17,25			10,6	7,33
	24,5	14,58			5,2	6,19
	23,3	16,01			4,9	4,46

Tabla 4.2: *Imágenes y valores asociados*

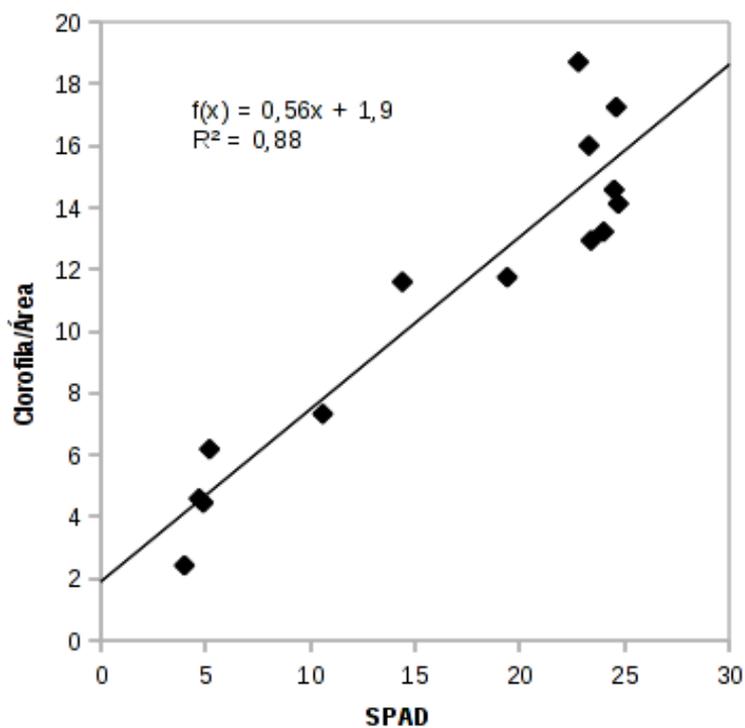


Figura 4.1: Diagrama de dispersión SPAD vs Contenido de clorofila (Espinacas en oscuridad)

Índice	r^2 Clorofila ($\mu\text{g}/\text{cm}^2$)	intervalo de confianza 95 %
R	$0,7686 \pm 0,1071$	[0,4375;0,9009]
G	$0,3027 \pm 0,1734$	[0,0123;0,6604]
B	$0,6876 \pm 0,1380$	[0,3330;0,8659]
$0,7582* R-B -0,1168* R-G +0,6414* G-B $	$0,5240 \pm 0,1791$	[0,0664;0,7950]
$(R-B)/(R+B)$	$0,5579 \pm 0,1767$	[0,1855;0,8329]
$R / (R+G+B)$	$0,3552 \pm 0,2189$	[0,0028;0,7373]
$G / (R+G+B)$	$0,7711 \pm 0,1024$	[0,4869;0,9244]

continua en la página siguiente

<i>continua de la página anterior</i>		
Índice	r^2 Clorofila ($\mu\text{g}/\text{cm}^2$)	intervalo de confianza 95 %
$B / (R+G+B)$	$0,6722 \pm 0,1355$	[0,2843;0,8605]
R/G	$0,7878 \pm 0,1011$	[0,4735;0,9099]
G/R	$0,7443 \pm 0,1146$	[0,3889;0,8974]
R - G	$0,7914 \pm 0,1027$	[0,3882;0,9162]
R - B	$0,2061 \pm 0,1781$	[0,0008;0,6544]
G - B	$0,7327 \pm 0,1061$	[0,3293;0,8702]
$(R-G)/(R+G)$	$0,7785 \pm 0,1056$	[0,4599;0,9023]
$(R-B)/(R+B)$	$0,5599 \pm 0,1709$	[0,1972;0,8540]
$(G-B)/(G+B)$	$0,7161 \pm 0,1318$	[0,3317;0,8830]
$(R-G)/(R+G+B)$	$0,7725 \pm 0,1057$	[0,3185;0,9113]
$(R-B)/(R+G+B)$	$0,4616 \pm 0,1926$	[0,0311;0,7733]
$(G-B)/(R+G+B)$	$0,7330 \pm 0,1198$	[0,3624;0,9042]
$(R-G)-(G-B)$	$0,7888 \pm 0,0942$	[0,4997;0,9087]
$((R-G)-(G-B))*(G-B)$	$0,7798 \pm 0,1059$	[0,4703;0,9164]
$((R-G)-(G-B))/(G-B)$	$0,7749 \pm 0,1172$	[0,2703;0,9130]
$ G-R +(2* R-B)$	$0,5811 \pm 0,1582$	[0,1201;0,8243]
H	$0,7551 \pm 0,1267$	[0,2139;0,9023]
S	$0,7119 \pm 0,1231$	[0,3177;0,8691]
V	$0,3105 \pm 0,1689$	[0,0135;0,6762]
$H/(H+S+V)$	$0,2951 \pm 0,2079$	[0,0006;0,6488]
$S/(H+S+V)$	$0,6552 \pm 0,1466$	[0,2634;0,8488]
$V/(H+S+V)$	$0,6945 \pm 0,1275$	[0,2523;0,8619]
H-S	$0,6622 \pm 0,1472$	[0,2962;0,8704]
<i>continua en la página siguiente</i>		

<i>continua de la página anterior</i>		
Índice	r^2 Clorofila ($\mu\text{g}/\text{cm}^2$)	intervalo de confianza 95 %
H-V	$0,6177 \pm 0,1539$	[0,2405;0,8530]
S-V	$0,6629 \pm 0,1453$	[0,2623;0,8672]
(H-S)/(H+S)	$0,4776 \pm 0,1672$	[0,0487;0,7168]
(H-V)/(H+V)	$0,7610 \pm 0,1052$	[0,3868;0,9017]
(S-V)/(S+V)	$0,6723 \pm 0,1281$	[0,2252;0,8475]
(H-S)/(H+S+V)	$0,6116 \pm 0,1523$	[0,1700;0,8366]
(H-V)/(H+S+V)	$0,7240 \pm 0,1183$	[0,3458;0,8871]
(S-V)/(H+S+V)	$0,6832 \pm 0,1323$	[0,1610;0,8523]

Tabla 4.3: *Valores de correlación para los distintos índices en las plantas de espinaca sometidas a oscuridad*

En la tabla 4.3 se muestra para cada índice la correlación obtenida mediante el valor r^2 , así como la desviación e intervalo de confianza, todos ellos una vez aplicada la técnica del bootstrapping. En ella puede observarse como los valores medios de el color rojo (R), tienen una relativamente alta correlación e intervalo de confianza, al contrario que los valores verdes (G), que la tienen muy baja. Estos resultados van en consonancia con los obtenidos por Mercado et al, (2010), para plantas de tomate. Siendo, incluso, en ambos resultados mejor la correlación del color rojo que del azul. Sin embargo, la correlación obtenida aquí es claramente peor que en el trabajo de Mercado.

En los siguientes gráficos se puede ver como no hay una buena correlación para dos de los índices citados en la introducción.

- El índice propuesto por Pagola et. al. [20] para la cebada, Ipca, no obtiene una buena correlación en este caso con las espinacas, con unos valores de $r^2 = 0,5240$, IC = [0.0664;0.7950]. ver figura 4.2.
- El índice propuesto por Kawashima [11], Ikaw, tampoco se ajusta para el caso de la espinaca, $r^2 = 0,5579$, IC = [0.1855;0.8329]. Figura 4.3.

En cambio para los índices en los que hay una relación entre los valores rojo (R) y verde (G), se ve una mayor correlación con los valores del SPAD. Tanto en forma racional como

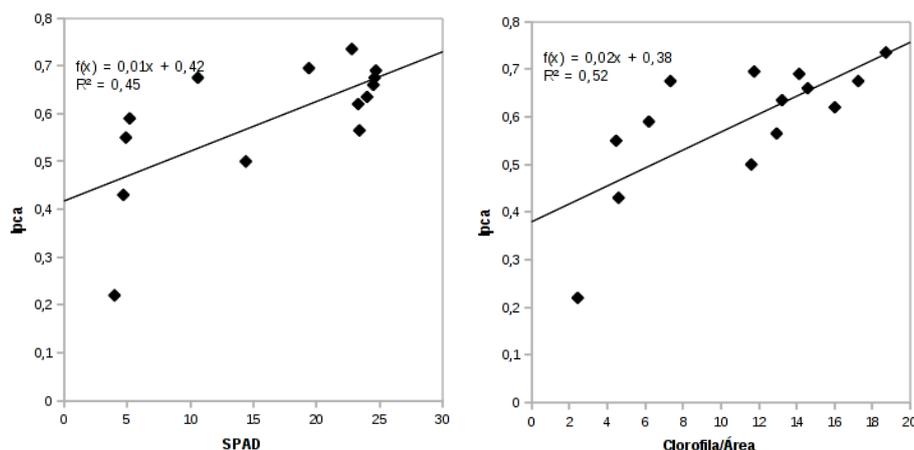


Figura 4.2: relaciones de I_{pca} vs SPAD y vs Clorofila/Área

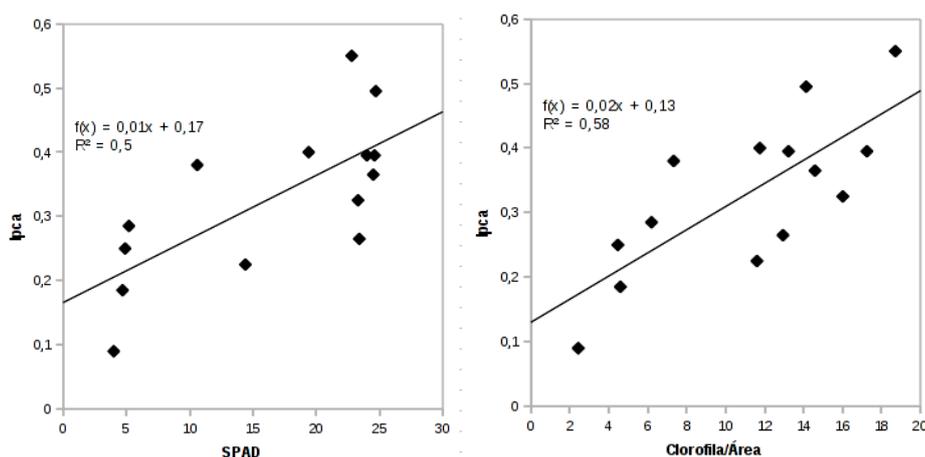


Figura 4.3: relaciones de I_{kaw} vs SPAD y vs Clorofila/Área

R/G y G/R, con un coeficiente de correlación lineal r^2 de 0,77 y 0,69 respectivamente y en unos intervalos de confianza de [0.437;0.900] y [0.333;0.865]; así como en los que la relación es de diferencia como $(R-G)/(R+G)$, $(R-G)/(R+G+B)$, $((R-G)-(G-B))/(G-B)$, $((R-G)-(G-B))*(G-B)$ y $(R-G)-(G-B)$ en todos los que r^2 es también del orden de 0,77.

- Índice $(R-G)-(G-B)$. Este índice obtiene una buena correlación con el contenido de clorofila en las hojas.

En concreto el r^2 obtenido es de 0.7888 ± 0.0942 , en un intervalo de confianza de [0.4997;0.9087].

- El índice R-G también presenta una alta correlación con la cantidad de clorofila presente en la hoja.

4.2. Índices obtenidos con el nivel medio de cada zona utilizando el algoritmo Watershed

En concreto el r^2 obtenido es de 0.7914 ± 0.1027 , en un intervalo de confianza de $[0.3882;0.9162]$.

- Este índice $\frac{R-G}{R+G}$ muestra una muy buena correlación con el contenido en clorofila.

En concreto el r^2 obtenido es de 0.7785 ± 0.1056 , en un intervalo de confianza de $[0.4599;0.9023]$.

En cualquier caso, ninguno de estos índices mejoran la fiabilidad ofrecida por el SPAD, aunque si que muestran que por medio del color se puede explicar en gran medida el contenido de clorofila de la hoja.

4.2. Índices obtenidos con el nivel medio de cada zona utilizando el algoritmo Watershed

Una vez aplicado el algoritmo Watershed y se calcularon los índices por regiones se obtuvieron los resultados que aparecen en la tabla 4.2.

Índice	r^2 Clorofila ($\mu\text{g}/\text{cm}^2$)	intervalo de confianza 95 %
R	$0,3748 \pm 0,2852$	$[0,0004;0,9012]$
G	$0,2012 \pm 0,2250$	$[0,0001;0,5542]$
B	$0,3646 \pm 0,2649$	$[0,0013;0,9573]$
$0.7582* R-B -0.1168* R-G +0.6414* G-B $	$0,4244 \pm 0,2608$	$[0,0002;0,8045]$
$(R-B)/(R+B)$	$0,3997 \pm 0,2511$	$[0,0001;0,7894]$
$R / (R+G+B)$	$0,4800 \pm 0,3338$	$[0,0002;0,3052]$
$G / (R+G+B)$	$0,3965 \pm 0,2895$	$[0,0009;0,9218]$
$B / (R+G+B)$	$0,3943 \pm 0,2583$	$[0,0012;0,9330]$
R/G	$0,4272 \pm 0,2755$	$[0,0006;0,9070]$
<i>continua en la página siguiente</i>		

4.2. Índices obtenidos con el nivel medio de cada zona utilizando el algoritmo Watershed

<i>continua de la página anterior</i>		
Índice	r ² Clorofila ($\mu\text{g}/\text{cm}^2$)	intervalo de confianza 95 %
G/R	0,3930 \pm 0,2852	[0,0005;0,8456]
R - G	0,4881 \pm 0,2640	[0,0000;0,7369]
R - B	0,2869 \pm 0,2972	[0,0000;0,5063]
G - B	0,4380 \pm 0,2916	[0,0030;0,9794]
(R-G)/(R+G)	0,3971 \pm 0,2787	[0,0001;0,8597]
(R-B)/(R+B)	0,4115 \pm 0,2544	[0,0000;0,8230]
(G-B)/(G+B)	0,4323 \pm 0,2773	[0,0029;0,9608]
(R-G)/(R+G+B)	0,3988 \pm 0,2894	[0,0016;0,9998]
(R-B)/(R+G+B)	0,3663 \pm 0,2428	[0,0001;0,6460]
(G-B)/(R+G+B)	0,4144 \pm 0,2792	[0,0036;0,9441]
(R-G)-(G-B)	0,4313 \pm 0,3054	[0,0006;0,9929]
(R-G)-(G-B))*(G-B)	0,4203 \pm 0,3055	[0,0004;0,9962]
((R-G)-(G-B))/(G-B)	0,5561 \pm 0,2836	[0,0069;0,4480]
G-R +(2* R-B)	0,4356 \pm 0,2640	[0,0068;0,9116]
H	0,5189 \pm 0,2848	[0,0022;0,9241]
S	0,4060 \pm 0,2739	[0,0042;0,9749]
V	0,2056 \pm 0,2310	[0,0000;0,5177]
H/(H+S+V)	0,2526 \pm 0,3021	[0,0000;0,3917]
S/(H+S+V)	0,3417 \pm 0,2571	[0,0007;0,9737]
V/(H+S+V)	0,3721 \pm 0,2771	[0,0003;0,9012]
H-S	0,3819 \pm 0,2722	[0,0012;0,9188]
H-V	0,3058 \pm 0,2613	[0,0001;0,6364]
S-V	0,3471 \pm 0,2555	[0,0001;0,8067]
<i>continua en la página siguiente</i>		

<i>continua de la página anterior</i>		
Índice	r^2 Clorofila ($\mu\text{g}/\text{cm}^2$)	intervalo de confianza 95 %
$(H-S)/(H+S)$	$0,3137 \pm 0,2721$	[0,0006;0,8476]
$(H-V)/(H+V)$	$0,4014 \pm 0,2763$	[0,0000;0,7199]
$(S-V)/(S+V)$	$0,3426 \pm 0,2689$	[0,0004;0,8537]
$(H-S)/(H+S+V)$	$0,3524 \pm 0,2687$	[0,0012;0,8836]
$(H-V)/(H+S+V)$	$0,3684 \pm 0,2831$	[0,0001;0,8800]
$(S-V)/(H+S+V)$	$0,2863 \pm 0,2714$	[0,0003;0,8327]

Tabla 4.4: *Valores de correlación para los distintos índices en las plantas de espinaca sometidas a oscuridad, después de aplicar el algoritmo watershed*

Como puede verse los índices obtenidos no solo no han mejorado, sino que han empeorado. Por tanto el esfuerzo computacional añadido no ha sido de ninguna utilidad.

El objetivo principal de segmentar las hojas en regiones y calcular los índices a partir de estas no era otro que el eliminar la influencia que podían tener aspectos como los nervios de las hojas que son blancos y perjudican el calculo del índice. En este trabajo se dan dos factores que pueden haber provocado que la calidad del índice no mejore al aplicar la segmentación.

- El primero de ellos es el pequeño tamaño muestral de que se dispone, para el que una pequeña desviación puede resultar fatal.
- Por otro lado, en las hojas de espinaca que se ha tenido en consideración, los nervios no tenían un papel tan importante, en lo relativo al área ocupada, como en otras hojas en los que su presencia era más clara. Esto último puede verse comparando las imágenes de la tabla 4.2, que son las que se han utilizado con este algoritmo y la imagen 3.11, que muestra claramente la presencia de nervios de color blanco.

4.3. Índices obtenidos mediante la programación genética

Para la ejecución del algoritmo genético solo se consideraron las 14 plantas de espinaca que habían obtenido unos resultados correctos en el laboratorio.

Para el tratamiento de los datos de evaluación del algoritmo genético se han promediado los valores de fitness obtenidos en las 20 repeticiones por configuración y el conjunto total de pruebas.

Estudiando la evolución del algoritmo respecto a su tamaño, número de individuos y la duración del mismo, en número de generaciones, se ve una muy ligera mejoría a medida que aumentan ambas, pero en ningún caso se obtienen resultados claramente superiores. En el tamaño de la población tampoco se encuentran diferencias entre los distintos valores estudiados (5,10,25,50,100,500), obteniéndose valores muy parecidos con una media de 0.647 y una desviación de 0.018. Los mejores valores obtenidos también presentan una desviación de 0.035 sobre el valor medio de 0.822.

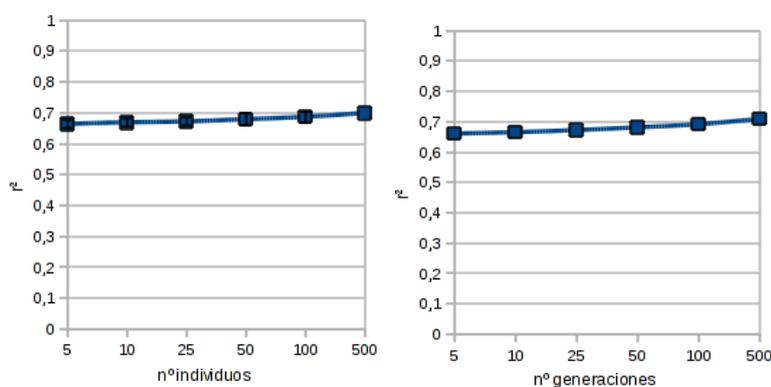


Figura 4.4: Tamaño y duración de la población

El análisis de los métodos de inicialización de la población, en la figura 4.5, (Fullinit, Growinit y Rampedinit) no ha mostrado diferencias entre ninguno de ellos, la media de sus valores medios ha sido de 0.648 con una desviación de 0.000016. Respecto a la media de sus mejores valores es .806 y la desviación .0015.

Respecto al parámetro de nivel dinámico, que permite o no el aumento del tamaño del individuo no se han encontrado diferencias entre habilitarlo o no. En todas las configuraciones se obtiene una media de fitness de 0.648 y una desviación muy baja de 0.000004. En el caso del mejor valor que se obtiene tampoco hay diferencias, una media de 0.821 y una desviación de 0.0000003. Figura 4.6.

En cuanto a los métodos de reemplazamiento las diferencias siguen sin apreciarse, para

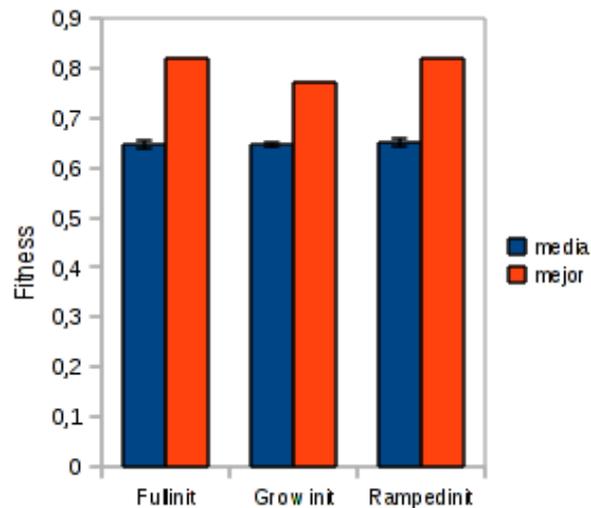


Figura 4.5: *Evaluación de los métodos de inicialización*

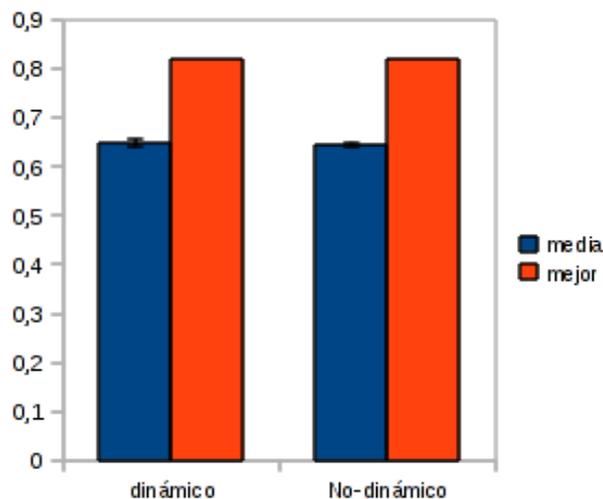


Figura 4.6: *Tamaño dinámico vs. no dinámico de los individuos*

los cuatro métodos estudiados (medio-elitismo, mantener el mejor, reemplazo y elitismo-total) la media de los valores medios sigue siendo de 0.646 y la desviación 0.000073. Mientras que para el caso de los mejores valores, la media es de 0.803 y la desviación 0.0017. Figura 4.7.

De los tres métodos de selección de padres, rank89, rank95 y absoluto, ninguno de ellos ha destacado sobre el resto. La media de los tres ha sido de 0.6708 y la desviación de 0.00613. Figura 4.8.

La selección de padres mediante los métodos de doble torneo, lexicTour, sus y torneo, no muestra diferencias, la media es de 0.648 con una desviación de 0.000027, y respecto

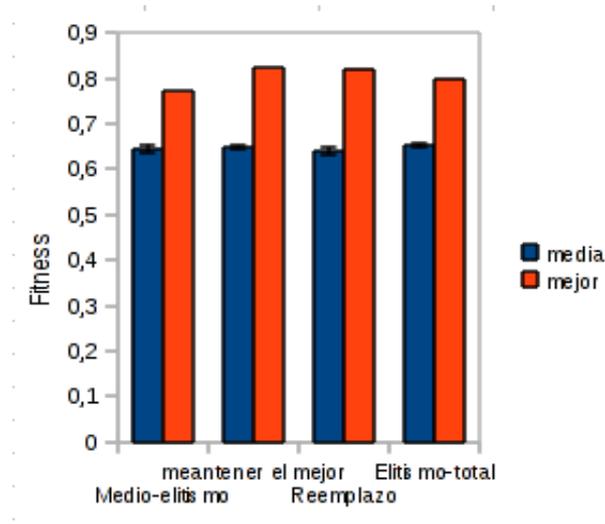


Figura 4.7: Evaluación de los métodos de reemplazo

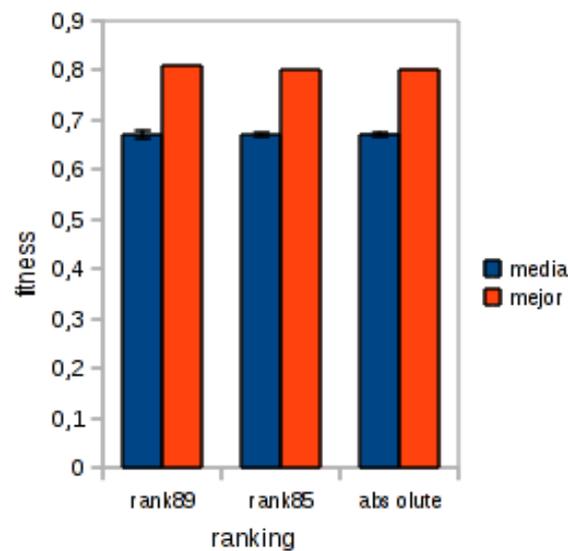


Figura 4.8: Evaluación de los métodos de ordenación de padres

a los mejores valores presentan una media de 0.794 con una desviación de 0.003. Figura 4.9.

Respecto a la comparación entre los métodos de evaluación del fitness regfitness y corrfitness En el caso de regfitness, el mejor fitness obtenido es de 37,092 que se corresponde a la función $\frac{G+B}{\frac{2 \cdot G}{B}}$ con un valor de correlación r^2 de 0.655, mientras que de todas la expresiones encontradas, la que mejor correlación muestra es $R \cdot \frac{R-G}{G^2}$, con un valor de 0.81678, pero que le corresponde un fitness de 157,25, muy alejado del mejor. En total, la

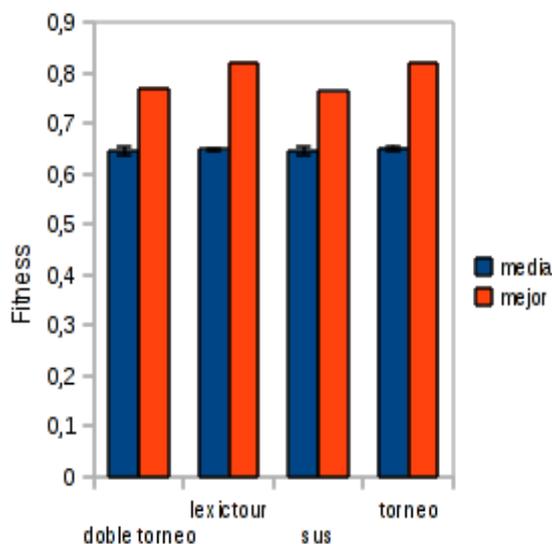


Figura 4.9: Evaluación de los métodos de selección de padres

media de los fitness obtenidos mediante la función de evaluación regfitness es de 106.52 con una desviación de 56,99. Y la media de las correlaciones correspondientes es de 0.5525 con una desviación de 0.17, alejada de la media de la función corrfitness que era de 0.645. Como se puede ver en la figura 4.10, no hay ninguna relación entre el valor calculado del fitness y la correlación del índice obtenido. La mayoría de los índices están en torno al valor 100 y se dan para la mayoría de los valores de correlación, formando casi una recta paralela al eje de la r^2 .

De las distintas ejecuciones realizadas para buscar una nueva relación entre los valores RGB de las imágenes y el contenido de clorofila de las hojas, se puede observar como siempre tiende a generar expresiones que dan como resultado valores de correlación en el entorno de 0,6.

Los mejores valores obtenidos utilizando los índices propuestos por otros autores, también presentan los valores en el mismo entorno de 0,8.

Después de todos los experimentos realizados, el algoritmo solo ha sido capaz de generar dos expresiones que mejoran, muy ligeramente, la relación que proporcionaban el resto de índices considerados. Para la evaluación de estas expresiones se ha utilizado el bootstrapping.

Como resultados se han obtenido los dos nuevos índices:

- Índice: $(R - G) \cdot \frac{R^3}{B}$ con un coeficiente de correlación $r^2 = 0,82701$. Figura 4.11.

Realizando el bootstrapping se obtienen los siguientes resultados: 0.8213 ± 0.0808 ,

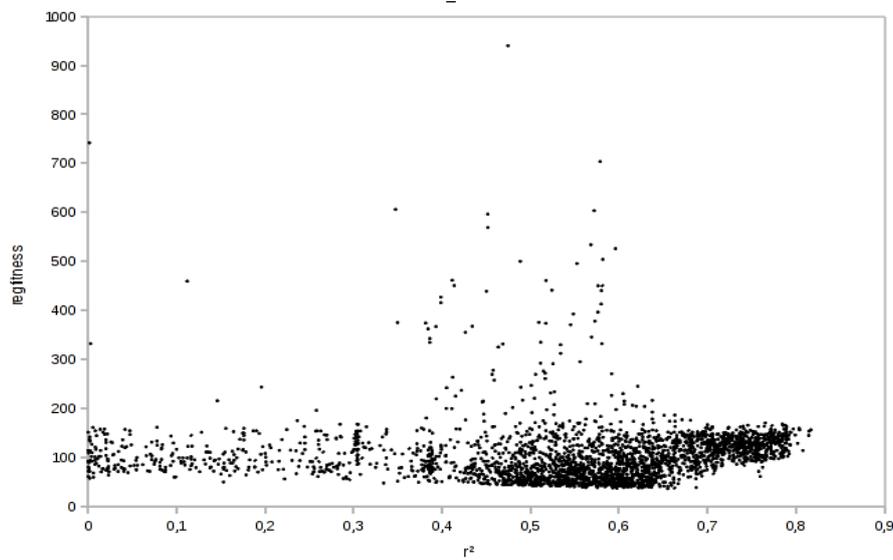


Figura 4.10: Relación entre el método de evaluación del fitness *regfitness* y la correlación obtenida por el índice

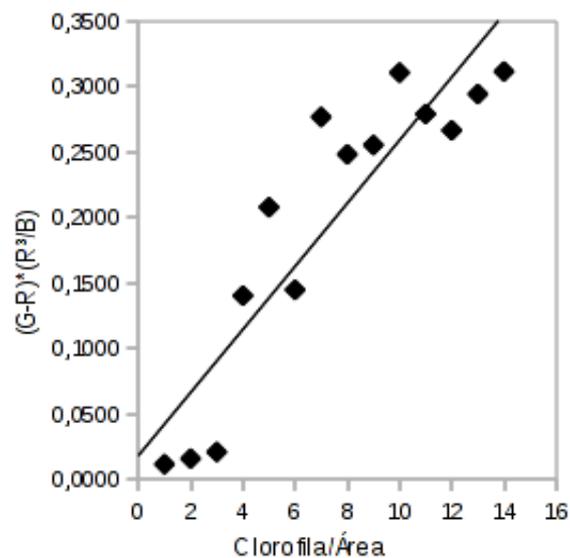


Figura 4.11: Relación entre el índice $(R - G) \cdot \frac{R^3}{B}$ y el contenido de clorofila

y el intervalo de confianza [0.4864;0.9238]

- Índice: $R^3 \cdot \frac{(2 \cdot G - R) \cdot (G - R)}{B}$ con un coeficiente de correlación $r^2 = 0,83784$. Figura 4.12. Realizando el bootstrapping se botienen los siguientes resultados: 0.8370 ± 0.0748 , y el intervalo de confianza [0.6011;0.9295]

Ambos índices mejoran ligeramente el valor de la correlación con el contenido de

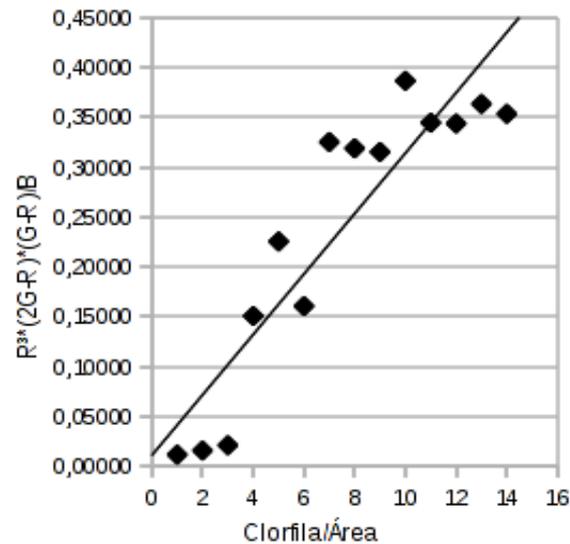


Figura 4.12: *Relacion entre el indice $R^3 \cdot \frac{(2G-R) \cdot (G-R)}{B}$ y el contenido de clorofila*

clorofila en las hojas, sin embargo esta mejora es muy leve y considerando que la cantidad de muestras ha sido baja, no se pueden establecer como definitivas.

4.4. Redes Neuronales

Para un entrenamiento de la red con una selección aleatoria de las entradas repartidas de forma 70 % entrenamiento, 15 % validación y 15 % comprobación; Con la función de evaluación de Levenberg-Marquardt y ajustando en función de la media de los errores absolutos. Se realizan 50 repeticiones para redes que varían desde 1 neurona hasta 20.

Como se puede ver en la figura 4.13, la media de las ejecuciones presenta un valor de correlación (r^2) muy bajo, y con una alta variabilidad. Si bien es cierto que puntualmente se obtienen configuraciones de la red con muy buenos resultados, superando el 0,9.

La situación anterior en la que solo unas pocas redes consiguen obtener buenos resultados es debido a que al tener pocos datos y a la inicialización aleatoria de la muestra para entrenar, hace que la red se estanque en mínimos locales, pero en algunas ejecuciones logra salir de los mismos y consigue unos buenos resultados.

Como resultado destacable, considero una red de 3 neuronas por su sencillez, ya que una red hasta con el doble de neuronas no obtiene mejores resultados. Una de las ejecuciones de esta red obtuvo los siguientes resultados una vez aplicada la técnica del bootstrapping.

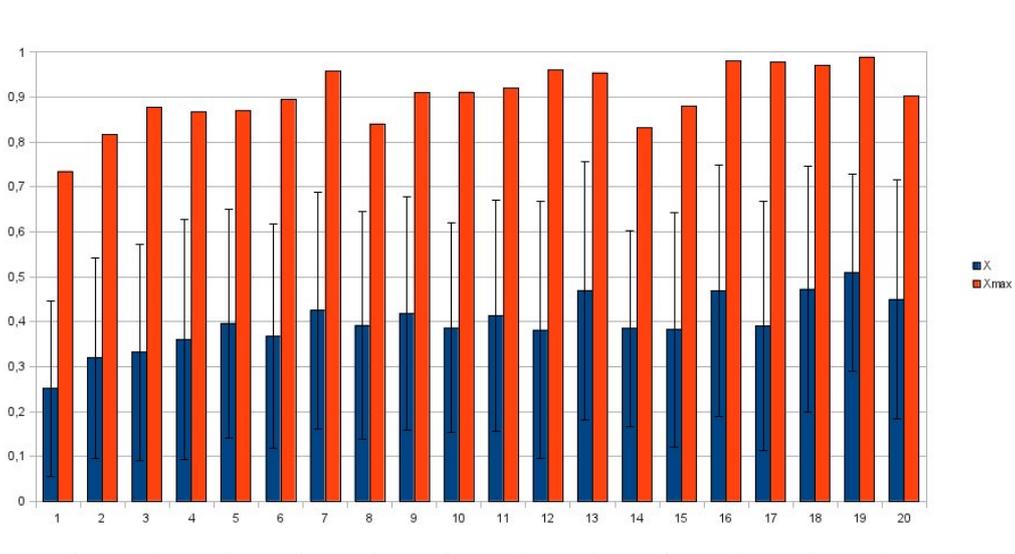


Figura 4.13: Valores medios, máximos y desviación de las 50 ejecuciones de la redes neuronales de una sola capa oculta

$r^2 = 0.8787 \pm 0.0731$ y un IC = [0.6195;0.9594]. Este resultado es el mejor resultado teniendo en cuenta todos los sistemas utilizados.

De todas formas, hay que considerar que esta red es valida únicamente para estos datos, ya que al ser la muestra muy pequeña no es conveniente hacer una extrapolación.

Si se añade más complejidad a la red por medio de una segunda capa oculta de neuronas, no se consigue mejorar el resultado de las redes con una sola capa, ver figura 4.14. Con un número de neuronas variable en cada capa entre 1 y 20, se obtienen unos valores solo ligeramente superiores a los de la red de una sola capa.

En ambos experimentos se aprecia que el número de neuronas no es determinante, excetpo en las capas con pocas neuronas (<3), el resto presentan resultados muy similares.

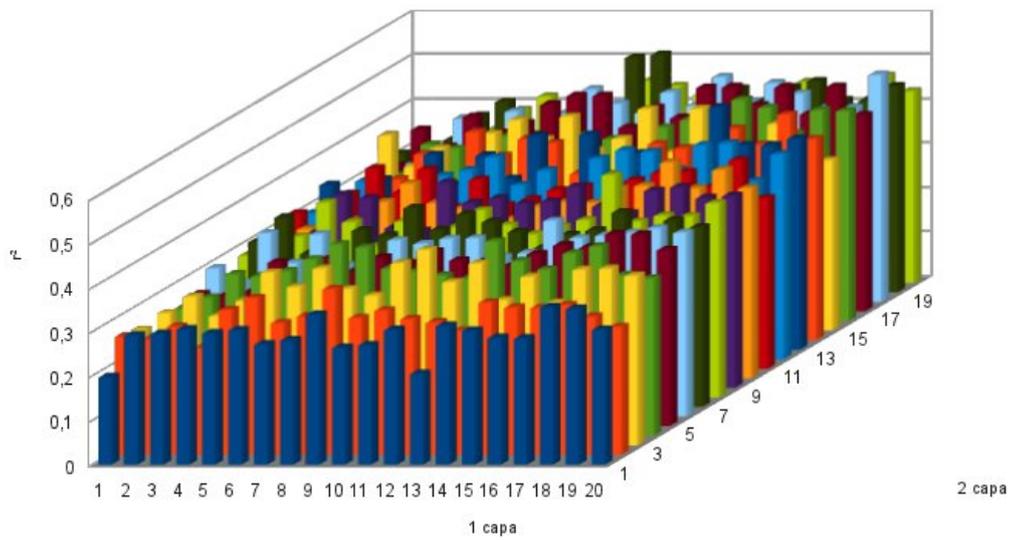


Figura 4.14: Valores medios de las 50 ejecuciones de la red neuronal, con 2 capas ocultas

Capítulo 5

Conclusiones

No se ha podido demostrar si los índices propuestos en la literatura eran correctos o adaptables a los casos de guisante y espinaca, por los problemas que hubo en el cultivo y como resultado el pequeño tamaño muestra que se ha visto reducido a 16 plantas de espinaca.

En lo que respecta a la utilización del algoritmo Watershed para segmentar la hoja y eliminar las partes que no tienen clorofila, tampoco se ha conseguido ninguna mejora, por el mismo motivo del pequeño tamaño muestra final, y que adicionalmente en esas plantas utilizables la presencia de nervios en la hoja era mínima.

En lo referente al desarrollo de una nueva metodología para estimar relaciones entre los valores RGB de los píxeles de una imagen de una hoja, y sus correspondiente contenido de clorofila, parece que se ha abierto una puerta que permita la obtención de nuevos índices mediante la programación genética. Si bien, como en los casos anteriores, dado el pequeño tamaño muestral los resultados obtenidos no se pueden considerar concluyentes.

Las red neuronal ha sido la técnica que mejores resultados ha producido, una simple red de una capa formada por tres neuronas ha sido capaz de obtener unos resultados muy buenos. Pero hay que tener en cuenta que dado el pequeño tamaño de la muestra esta red es valida solo para esta misma muestra.

Como líneas a futuro hay que volver a realizar los experimentos, asegurando unas muestras mas abundantes y representativas, para poder verificar mejor la validez de las metodologias de programación genética y sobre todo de las redes neuronales.

ANEXOS

Anexo: Código fuente

Los archivos utilizados para desarrollar este trabajo han sido los siguientes:

- `recortar.m`

En este archivo se encuentra el código encargado de recortar la imagen original y dejar únicamente la parte que interesa, que es la hoja.

- `indicesRGB.m` e `indicesHSV.m`

En estos archivos está el código que a partir de la imagen de la hoja genera todos los índices que se han considerado en la primera parte de este trabajo.

- `minimos.m` e `inundacion.m`

El primero de estos dos archivos genera los mínimos a utilizar en la segmentación Watershed. Mientras que el segundo a partir de los mínimos generados por el primero procede a la creación de las regiones Watershed.

- `water.m`

Este archivo invoca al procedimiento inundación y obtiene como respuesta una imagen con las 20-30 regiones Watershed. A continuación calcula cada uno de los índices para cada región y los une mediante una suma ponderada según el tamaño de la región.

- `genetico.m`

Este archivo configura el algoritmo implementado en `gplab.toolbox` y lo ejecuta con los parámetros seleccionados.

- `corrfitness.m`

Función de evaluación del fitness según la correlación entre las dos variables.

- `redneuronal.m`

Archivo en el que se implementa la evaluación de las redes neuronales.

```
1 function recorte = recortar (imagen)
2 %funcion que recorta de la imagen todo aquello
3 %que no sea una hoja
4
5 %Componentes RGB
6 R=imagen(:,:,1);
7 G=imagen(:,:,2);
8 B=imagen(:,:,3);
9
10 %Filtro verdor
11 X=(abs(double(R)-double(G))-abs(double(G)-double(B)))<5;
12
13 %Filtro de carne
14 Y=abs(double(G)-double(B))>40;
15
16 %Combinacion de los fitlros
17 R=X .* Y;
18
19 S = imclearborder(R);
20
21 J=bwareaopen(S,25000);%valor arbitrario, despues de algunas pruebas
22
23 J=imfill(J,'holes');
24
25 s=regionprops(J,'perimeter','area','Orientation');
26
27
28 index = 1;
29 if length(s) > 1
30     for i = 2:length(s)
31         if ((s(i).Perimeter^2/s(i).Area) < (s(i-1).Perimeter^2/s(i-1).Area) && (s(i).
32             index = i;
33         end
34     end
35 end
36
37
38 STATS = regionprops(J,'Orientation');
39
40 J = imrotate(J,-STATS(index).Orientation);
41
42 imagen = imrotate(imagen,-STATS(index).Orientation);
```

```

43
44 STATS = regionprops(J, 'BoundingBox');
45
46 imagencortada=imcrop(imagen, [STATS(index).BoundingBox(1) STATS(index).BoundingBox(2)
47
48
49 dimensiones=size(imagencortada);
50 longitud=dimensiones(2);
51 altura=dimensiones(1);
52 imagencortada=imcrop(imagencortada, [longitud*0.075 altura*0.075 longitud*0.85 altura*
53
54 recorte = imagencortada;

```

```

1 function vectorIndices = indicesRGB (R,G,B)
2 vectorIndices(1) = R;
3 vectorIndices(2) = G;
4 vectorIndices(3) = B;
5 vectorIndices(4)=0.7582*abs(R-B)-0.1168*abs(R-G)+0.6414*abs(G-B);
6 vectorIndices(5)=(R-B)/(R+B);
7 vectorIndices(6) = R / (R+G+B);
8 vectorIndices(7) = G / (R+G+B);
9 vectorIndices(8) = B / (R+G+B);
10 vectorIndices(9) = R/G;
11 vectorIndices(10) = G/R;
12 vectorIndices(11) = R - G;
13 vectorIndices(12) = R - B;
14 vectorIndices(13) = G - B;
15 vectorIndices(14) = (R-G)/(R+G);
16 vectorIndices(15) = (R-B)/(R+B);
17 vectorIndices(16) = (G-B)/(G+B);
18 vectorIndices(17) = (R-G)/(R+G+B);
19 vectorIndices(18) = (R-B)/(R+G+B);
20 vectorIndices(19) = (G-B)/(R+G+B);
21 vectorIndices(20) = (R-G)-(G-B);
22 vectorIndices(21) = ((R-G)-(G-B))*(G-B);
23 vectorIndices(22) = ((R-G)-(G-B))/(G-B);
24 vectorIndices(23) = abs(G-R)+(2*abs(R-B));

```

```

1 function vectorIndices = indicesHSV (H,S,V)
2 vectorIndices(1) = H;

```

```
3 vectorIndices(2) = S;
4 vectorIndices(3) = V;
5 vectorIndices(4) = H / (H+S+V);
6 vectorIndices(5) = S / (H+S+V);
7 vectorIndices(6) = V / (H+S+V);
8 vectorIndices(7) = H - S;
9 vectorIndices(8) = H - V;
10 vectorIndices(9) = S - V;
11 vectorIndices(10) = (H-S)/(H+S);
12 vectorIndices(11) = (H-V)/(H+V);
13 vectorIndices(12) = (S-V)/(S+V);
14 vectorIndices(13) = (H-S)/(H+S+V);
15 vectorIndices(14) = (H-V)/(H+S+V);
```

```
1 function semillas=minimos(A)
2 %Se calculan las semillas
3
4 F=A(:, :, 2);
5 [fil, col]=size(F);
6 %invertimos la imagen
7 F=abs(double(F)-255);
8 %cogemos G<F
9 G=double(F)-1;
10 %reconstruimos F con G con 8-vecinos
11 I=imreconstruct(G,F,8);
12 %en la diferencia algebraica tenemos los minimos
13 I=F-I;
14 s= strel('square',2);
15 %cerramos y juntamos pixels cercanos
16 I=imclose(I,s);
17 %eliminamos pixels sueltos (minimos con poco peso)
18 I = imerode(I,s);
19 %dilatamos para tener las zonas unidas
20 I = imdilate(I, s);
21 %Bucle en el que se va adelgazando hasta obtener puntos
22 vecino=1;
23 while vecino~=0
24     vecino=0;
25 I=bwmorph(I, 'shrink');
26 for i=1:fil
27     for j=1:col
```

```
28     if I(i,j)~=0
29         if i>1 && j>1 && I(i-1,j-1)~=0
30             vecino=vecino+1;
31         end
32         if i>1 && I(i-1,j)~=0
33             vecino=vecino+1;
34         end
35         if i>1 && j<col && I(i-1,j+1)~=0
36             vecino=vecino+1;
37         end
38         if j<col && I(i,j+1)~=0
39             vecino=vecino+1;
40         end
41         if i<fil && j<col && I(i+1,j+1)~=0
42             vecino=vecino+1;
43         end
44         if i<fil && I(i+1,j)~=0
45             vecino=vecino+1;
46         end
47         if i<fil && j>1 && I(i+1,j-1)~=0
48             vecino=vecino+1;
49         end
50         if j>1 && I(i,j-1)~=0
51             vecino=vecino+1;
52         end
53     end
54 end
55 end
56 end
57
58 h=1;
59 k=1;
60 %llenamos semillas con los minimos
61 for i=1:fil
62     for j=1:col
63         if I(i,j)~=0
64             semillas(h,1)=k;
65             semillas(h,2)=i;
66             semillas(h,3)=j;
67             semillas(h,4)=A(i,j,2);
68             h=h+1;
69         end
70     end
    k=k+1;
```

```
71     end
72 end
```

```
1  function A2 = inundacion(A)
2  %Este es el procedimiento de llenado o inundacion del watershed basado en
3  %una serie de buffers o vectores
4
5
6  fin=0;
7  [F,C,Z]=size(A);
8  buffer=zeros(256,1,4);
9  B=zeros(F,C,2);
10 B(:, :, 1)=A(:, :, 2);
11
12 semillas=minimos(A);
13 [F2,C2]=size(semillas);
14 %metemos las semillas en sus buffers correspondientes segun valor
15 for i=1:F2
16     B(semillas(i,2), semillas(i,3), 2)=semillas(i,1);
17     for j=1:4
18         buffer(semillas(i,4)+1, 2, j)=0;
19         buffer(semillas(i,4)+1, 1, j)=semillas(i, j);
20     end
21 end
22 %bucle de llenado
23 while fin==0
24 i=1;
25 %cogemos desde el mas viejo al nuevo el primer buffer con algun valor
26 while buffer(i,1,1)==0
27     if i+1==257
28         fin=1;
29         break;
30     end
31     i=i+1;
32 end
33 %si el buffer ya no tiene elementos hemos terminado
34 if fin==1
35     break;
36 end
37 %Cogemos las coordenadas de los pixels vecinos controlando que si estamos
38 %en los bordes no nos vayamos fuera de la matriz
```

```

39 if (buffer(i,1,2)-1)>=1
40     vecinos(1,:)=[buffer(i,1,2)-1,buffer(i,1,3)];
41 else
42     vecinos(1,:)=[buffer(i,1,2),buffer(i,1,3)];
43 end
44 if (buffer(i,1,3)+1)<=C
45     vecinos(2,:)=[buffer(i,1,2),buffer(i,1,3)+1];
46 else
47     vecinos(2,:)=[buffer(i,1,2),buffer(i,1,3)];
48 end
49 if (buffer(i,1,2)+1)<=F
50     vecinos(3,:)=[buffer(i,1,2)+1,buffer(i,1,3)];
51 else
52     vecinos(3,:)=[buffer(i,1,2),buffer(i,1,3)];
53 end
54 if (buffer(i,1,3)-1)>=1
55     vecinos(4,:)=[buffer(i,1,2),buffer(i,1,3)-1];
56 else
57     vecinos(4,:)=[buffer(i,1,2),buffer(i,1,3)];
58 end
59 %Bucle en el que si el pixel analizado no tiene Id, le damos la del
60 %elemento que estamos mirando y lo encolamos en su correspondiente buffer
61 for j=1:4
62     %Si no tiene ID lo tratamos sino no
63     if B(vecinos(j,1),vecinos(j,2),2)==0
64         %Si el buffer no tiene elementos lo metemos como primer elemento
65         if buffer(B(vecinos(j,1),vecinos(j,2),1),1,1)==0
66             B(vecinos(j,1),vecinos(j,2),2)=buffer(i,1,1);
67             buffer(B(vecinos(j,1),vecinos(j,2),1),1,1)=buffer(i,1,1);
68             buffer(B(vecinos(j,1),vecinos(j,2),1),1,2)=vecinos(j,1);
69             buffer(B(vecinos(j,1),vecinos(j,2),1),1,3)=vecinos(j,2);
70             buffer(B(vecinos(j,1),vecinos(j,2),1),1,4)=B(vecinos(j,1),vecinos(j,2),1)
71         %Sino tenemos que encolarlo moviendo el resto
72     else
73         h=2;
74         while buffer(B(vecinos(j,1),vecinos(j,2),1),h,1)~=0
75             h=h+1;
76         end
77         %despues de mover el resto lo encolamos
78         B(vecinos(j,1),vecinos(j,2),2)=buffer(i,1,1);
79         buffer(B(vecinos(j,1),vecinos(j,2),1),h,1)=buffer(i,1,1);
80         buffer(B(vecinos(j,1),vecinos(j,2),1),h,2)=vecinos(j,1);
81         buffer(B(vecinos(j,1),vecinos(j,2),1),h,3)=vecinos(j,2);

```

```

82         buffer(B(vecinos(j,1),vecinos(j,2),1),h,4)=B(vecinos(j,1),vecinos(j,2),1)
83         for m=1:4
84             buffer(B(vecinos(j,1),vecinos(j,2),1),h+1,m)=0;
85         end
86     end
87 end
88 end
89 h=1;
90 while buffer(i,h,1)~=0
91     for m=1:4
92         buffer(i,h,m)=buffer(i,h+1,m);
93     end
94     h=h+1;
95 end
96 end
97 %Fin bucle llenado
98
99 %segun las Ids pintamos de nuevo la imagen
100 for i=1:F
101     for j=1:C
102         for k=1:F2
103             if B(i,j,2)==semillas(k,1)
104                 A2(i,j,1)=A(semillas(k,2),semillas(k,3),1);
105                 A2(i,j,2)=A(semillas(k,2),semillas(k,3),2);
106                 A2(i,j,3)=A(semillas(k,2),semillas(k,3),3);
107             else
108                 if B(i,j,2)==0
109                     A2(i,j,1)=A(i,j,1);
110                     A2(i,j,2)=A(i,j,2);
111                     A2(i,j,3)=A(i,j,3);
112                 end
113             end
114         end
115     end
116 end

```

```

1 function indice = water(IMG)
2 IMGBW1 = rgb2gray(IMG);
3
4
5 se = strel('ball',75,75);

```

```

6  IMGBW=imerode(IMGBW1,se);
7  IMGBW=imdilate(IMGBW,se);
8  L = inundacion(IMG);
9
10 L=rgb2gray(L);
11 pixels=numel(IMGBW)/3;
12 valores=unique(L);
13 m=length(valores);
14 m=m(1);
15 indice=0;
16 for j=1:m           %calcular la media de cada region
17     DL=L(:, :, 1)==valores(j);
18     sumwater=sum(sum(DL~=0));
19     DL2 = uint8(DL);
20     IMG2(:, :, 1) = IMG(:, :, 1) .* DL2;
21     IMG2(:, :, 2) = IMG(:, :, 2) .* DL2;
22     IMG2(:, :, 3) = IMG(:, :, 3) .* DL2;
23     %media de los valores de esa region waters
24     R=sum(sum(IMG2(:, :, 1)))/sumwater;
25     G=sum(sum(IMG2(:, :, 2)))/sumwater;
26     B=sum(sum(IMG2(:, :, 3)))/sumwater;
27
28     indice=indice+indicesRGB(R,G,B)*sumwater/pixels;
29
30 end;

```

```

1  pathr = '/home/asier/data/Proyecto/resultados/';
2  fichero = (strcat(pathr,'genetico.csv'));
3  salida = fopen(fichero, 'w');
4
5  inprobttype={'fixed' 'variable'};
6  dynamiclevel=[0,1];
7  indynlev=[0.75,1];
8  inmaxlev=[0.5,0.75,1];
9  realmxlev=[20];
10 elitism={'replace' 'keepbest' 'halfelitism' 'totalelitism'};
11 depthnodes={'1' '2'};
12 initpop={'fullinit' 'growinit' 'rampedinit'};
13 samp={'doubletour' 'tournament' 'sus' 'lexictour'};
14 survival={'fixedpopsize' 'resources' 'pivotfixe'};
15 maxpop=[5,10,25,50,100];

```

```
16     maxgen=[5,10,25,50,100];
17
18     cabeceras = strcat('dynamiclevel;realmxlevel;indynlev;inmaxlev;inproptype;elitism');
19     fprintf(salida,cabeceras);
20
21     for k=maxgen
22     for j=maxpop
23     for rml=realmxlev
24     for idl=indynlev
25     for iml=inmaxlev
26     for dl=dynamiclevel
27     for ipt=inproptype
28     for e=elitism
29     for dn=depthnodes
30     for l=initpop
31         for i=samp
32             for s=survival
33
34
35                 if char(dn)=='1' rml2 = round((rml/5)+1);
36
37                 variables = char(strcat(sprintf('%4.6f;',dl),sprintf('%4.6f;',rml2)));
38
39                 fprintf(salida,variables);
40
41
42                 for rep=1:10
43
44
45                     p=resetparams;
46
47                     p=setoperators(p,'crossover',2,2,'mutation',1,1);
48                     p.operatorprobstype=char(ipt);
49
50                     p.minprob=0.05;
51
52                     p = setfunctions(p,'plus',2,'minus',2,'times',2,'mydivide',2);
53
54
55                     p.datafilex='/home/asier/data/Proyecto/resultados/esposcrgb2.';
56                     p.datafiley='/home/asier/data/Proyecto/resultados/espOscClAr.';
57
58                     p.initpoptype=char(l);
```

```
59         p.elitism=char(e);
60         p.sampling = char(i);
61         p.tournamentsize = 4;
62         p.calcdiversity={'uniquegen'};
63         p.survival=char(s);
64         %p.calccomplexity=1;
65
66         p.dynamiclevel = dl;
67
68         if dl== 0
69             p.fixedlevel = 1;
70         else
71             p.fixedlevel = 0;
72         end
73
74         if char(dn)=='1'
75             p.realmaxlevel = rml2;
76         else
77             p.realmaxlevel = rml;
78         end
79
80         p.inicmaxlevel = round(p.realmaxlevel*iml);
81         p.inicdynlevel = round(p.realmaxlevel*idl);
82         p.depthnodes=char(dn);
83         p=setparams(p, 'calcfitness=asierfitness');
84
85         warning('off', 'all');
86
87         [v,b]=gplab(j,k,p);
88
89
90         fprintf(salida, strcat(sprintf('%4.6f;',b.fitness),b.str, ';'))
91
92
93         end
94         fprintf(salida, ';\n');
95     end
96 end
97 end
98 end
99 end
100 end
101 end
```

```

102     end
103
104     end
105     end
106     end
107     end
108     end
109     fclose(salida);

```

```

1  function ind=corrfitness(ind,params,data,terminals,varsvals)
2  %ASIERFITNESS    Measures the fitness of a GPLAB individual.
3  % ASIERFITNESS(INDIVIDUAL,PARAMS,DATA,TERMINALS,VARSVALS) returns
4  % the fitness of INDIVIDUAL, measured as the sum of differences
5  % between the obtained and expected results, on DATA dataset, and
6  % also returns the result obtained in each fitness case.
7  %
8  % Input arguments:
9  %     INDIVIDUAL - the individual whose fitness is to measure (struct)
10 %     PARAMS - the current running parameters (struct)
11 %     DATA - the dataset on which to measure the fitness (struct)
12 %     TERMINALS - the variables to set with the input dataset (cell array)
13 %     VARSVALS - the string of the variables of the fitness cases (string)
14 % Output arguments:
15 %     INDIVIDUAL - the individual whose fitness was measured (struct)
16 %
17 % See also CALCFITNESS, ANTFITNESS, REGFITNESS
18 %
19 % Copyright (C) 2003-2007 Sara Silva (sara@dei.uc.pt)
20 % Acknowledgements: Marco Medori (marco.medori@poste.it) and Bruno Morelli
21 % This file is part of the GPLAB Toolbox
22
23 X=data.example;
24 outstr=ind.str;
25 for i=params.numvars:-1:1
26     outstr=strrep(outstr, strcat('X', num2str(i)), strcat('X(:,', num2str(i), ')'));
27 end
28 %outstr
29 try
30     res=eval(outstr);
31 catch
32     % because of the "nesting 32" error of matlab

```

```

33     res=str2num(evaluate_tree(ind.tree,X));
34 end
35
36 % if the individual is just a terminal, res is just a scalar, but we want a vector:
37 if length(res)<length(data.result)
38     res=res*ones(length(data.result),1);
39 end
40
41 r = corr(res,data.result);
42 rsq=r^2;
43
44 ind.result=res;
45
46 % raw fitness:
47 ind.fitness=-rsq; %lower fitness means better individual rsq always +, and better near
48 % now limit fitness precision, to eliminate rounding error problem:
49 ind.fitness=fixdec(ind.fitness,params.precision);

```

```

1 pathr = '/home/asier/data/Proyecto/resultados/';
2 fichero = (strcat(pathr,'rneuronal2.csv'));
3 salida = fopen(fichero, 'w');
4
5 fprintf(salida,'numero neuronas;resultados\n');
6
7 for i=1:20
8     for j=1:20
9         n=char(sprintf('%4.6f;',i));
10        fprintf(salida,n);
11
12        for k=1:50
13
14
15
16 x=importdata('/home/asier/data/Proyecto/resultados/esposcrgb2.csv','\t',0);
17 t=importdata('/home/asier/data/Proyecto/resultados/espOscClAr.csv','\t',0);
18 net = feedforwardnet([i,j]);
19 net = train(net,transpose(x),transpose(t));
20 y=net(transpose(x));
21 perf = perform(net,y,transpose(t));
22 p = regstats(transpose(t),y,'linear');
23

```

```
24 fprintf(salida,sprintf('%4.6f;',p.rsquare^2));
25     end
26     fprintf(salida,'\n');
27     end
28 end
29
30 plot(transpose(t),y,'o');
```

Bibliografía

- [1] *GPLAB-a genetic programming toolbox for MATLAB.*
- [2] Mercado-Luna A., Rico-García E., Lara-Herrera A., Soto-Zarazúa G., Ocampo-Velázquez R., Guevara-González R., Herrera-Ruiz G., and Torres-Pacheco I. Nitrogen determination on tomato (*lycopersicon esculentum* mill.) seedlings by color image analysis (rgb). *African Journal of Biotechnology*, 9(33):5326–5332, 2010.
- [3] A. Alvira. Comparación de métodos de determinación de clorofila en plantas con relaciones hídricas alteradas. Proyecto fin de máster de la UPNA., 2010.
- [4] M. A. Atta-Aly. Effect of nickel addition on the yield and quality of parsley leaves. *Scientia Horticulturae*, 82(1-2):9–24, 1999.
- [5] S. Barro and J. Mira. *Computación neuronal*. Cursos e Congresos Da Universidade de Santiago de Compostela Series. Universidade de Santiago de Compostela, Servicio de Publicacións e Intercambio Científico, 1995.
- [6] Bradley Efron. *The Jackknife, the bootstrap and other resampling plans*. Number 38 in Regional Conference Series in applied mathematics. Society for Industrial and applied mathematics, 1982.
- [7] A. E. Eiben and J. E. Smith. *Introduction to Evolutionary Computing*. 2^a edición. Springer, 2007.
- [8] P. Garcia. Evaluación de métodos de determinación de clorofila en plantas con relaciones hídricas alteradas. Proyecto fin de carrera de la UPNA., 2000.
- [9] Hao Hu, He qin Liu, Hao Zhang, Jing huan Zhu, Xu guo Yao, Xiao bin Zhang, and Ke feng Zheng. Assessment of chlorophyll content based on image color analysis, comparison with spad-502. *2010 2nd International Conference on Information Engineering and Computer Science (ICIECS)*, (1-3), 2010.

-
- [10] Nahida Jelali, Imen Ben Salah, Wissal M'sehli, Silvia Donnini, Graziano Zocchi, and Mohamed Gharsalli. Comparison of three pea cultivars (*pisum sativum*) regarding their responses to direct and bicarbonate-induced iron deficiency. *Scientia Horticulturae*, 129(4):548–553, 2011.
- [11] S. Kawashima and M. Nakatani. An algorithm for estimating chlorophyll content in leaves using a video camera. *Annals of Botany*, (81):49–54, 1998.
- [12] J. R. Koza. Genetic programming.
- [13] Yan-Ju Liu, Yi-Ping Tong, Yong-Guan Zhu, Hui Ding, and F. Andrew Smith. Leaf chlorophyll readings as an indicator for spinach yield and nutritional quality with different nitrogen fertilizer applications. *Journal of Plant Nutrition*, 29(7):1207–1217, 2006.
- [14] Minolta Camera Co. Ltd. *Chlorophyll meter SPAD-502. Instruction Manual.*, 1989.
- [15] K. Mengel and E. A. Kirkby. *Principles of plant nutrition*. International Potash Institute, Bern, 1987.
- [16] O.A. Monje and B. Bugbee. Inherent limitation of nondestructive chlorophyll meters: a comparison of tow types of meters. *Hortoscience*, (27(1)):69–71, 1992.
- [17] A. Mozafar. Nitrogen fertilizers and the amount of vitamins in plants: A review. *Journal of Plant Nutrition*, (16):2479–2506, 1993.
- [18] B. Olav Hoel and K. Asbjorn Solhaug. Effect of irradiance on chlorophyll estimation with the minolta spad-502 leaf chlorhyll meter. *Annals of Botany*, (82):389–392, 1998.
- [19] Steering Group on Food Surveillance. Working Party on Nitrate, Related Compounds in Food, Great Britain. Working Party on Nitrate, Related Compounds in Food, Fisheries Great Britain. Ministry of Agriculture, and Food. *Nitrate, nitrite and N-nitroso compounds in food: the twentieth report of the Steering Group on Food Surveillance, the Working Party on Nitrate and Related Compounds in Food*. Food surveillance paper. H.M.S.O., 1987.
- [20] M. Pagola, R. Ortiz, I. Irigoyen, H. Bustince, E. Barrenechea, P. Aparicio-Tejo, C. Lamsfus, and B. Lasa. New method to assess barley nitrogen nutrition status based on imagee colour analysis . comparision with spad-502. *Computers and electronic in agriculture*, (65):213–218, 2009.

-
- [21] S. Peng, F. V. Garcia, R. C. Laza, A. L. Sanico, R. M. Visperas, and K. G. Cassman. Increased n-use efficiency using a chlorophyll meter on high yielding irrigated rice. *Field Crops Research*, (47):243–252, 1996.
- [22] W.P. Piekielek, R.H. Fox, J.D. Toth, and K.E. McNeal. Use of a chlorophyll meter at the early dent stage of corn to evaluate nitrogen sufficiency. *Agronomy Journal*, (87 (3)):403–408, 1995.
- [23] Thaiparnit S. and Srinonchat J. Apply image processing technique to determine the correlation of chlorophyll. *6th International Conference on Electrical Engineering/-Electronics, Computer, Telecommunications and Information Technology*, 2:1116 – 1119, 2009.
- [24] Leila Shafarenko, Maria Petrou, and Josef Kittler. Automatic watershed segmentation of randomly textured color images. *IEEE Transactions on Image Processing*, 6(11):1530–1544, 1997.
- [25] M Tabeke, T Yoneyama, K Inada, and T Murakami. Spectral reflectance ratio of rice canopy for estimating cop nitrogen status. *Plant and Soil*, (122):295–297, 1990.
- [26] I. Terashima and T. Saeki. Light environment within a leaf i. optical properties of paradermal sections of camellia leaves with special reference to differences in the optical properties of palisade and spongy tissues. *Plant and Cell Physiology*, (24):1493–1501, 1983.
- [27] F.T. Turner and M.F. Jund. Chlorophyll meter to predict nitrogen topdress requirement for semidwarf rice. *Agronomy Journal*, (83(5)):926–928, 1991.
- [28] Satya Prakash Yadav, Yasuomi Ibaraki, and S. Dutta Gupta. Estimation of the chlorophyll content of micropropagated potato plants using rgb based image analysis. *Plant Cell, Tissue and Organ Culture (PCTOC) Journal of Plant Biotechnology*, (100):183–188, 2009.