

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS
INDUSTRIALES Y DE TELECOMUNICACIÓN

UNIVERSIDAD DE CANTABRIA



Trabajo Fin de Grado

**DEEP LEARNING APLICADO AL
DIAGNÓSTICO DE SOLDADURA MEDIANTE
ESPECTROSCOPIA ÓPTICA DE PLASMAS**

**(Deep Learning applied to Welding
Diagnostics via Plasma Optical Spectroscopy)**

Para acceder al Título de

Graduado en Ingeniería de Tecnologías de Telecomunicación

Autor: César Abascal Gutiérrez

Octubre – 2017

GRADUADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

CALIFICACIÓN DEL TRABAJO FIN DE GRADO

Realizado por: César Abascal Gutiérrez

Director del TFG: Jesús María Mirapeix Serrano

Título: “*Deep Learning* aplicado al análisis de soldadura mediante espectroscopía óptica de plasmas”

Title: “Deep Learning applied to Welding Diagnostics via Plasma Optical Spectroscopy “

Presentado a examen el día:

Para acceder al Título de

GRADUADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

Composición del Tribunal:

Presidente (Apellidos, Nombre): María Ángeles Quintela Incera

Secretario (Apellidos, Nombre): Héctor Posadas Cobo

Vocal (Apellidos, Nombre): Jesús María Mirapeix Serrano

Este Tribunal ha resuelto otorgar la calificación de:

Fdo.: El Presidente

Fdo.: El Secretario

Fdo.: El Vocal

Fdo.: El Director del TFG
(Sólo si es distinto del Secretario)

Vº Bº del Subdirector

Trabajo Fin de Grado N°
(A asignar por Secretaría)

ÍNDICE

| | |
|---|-----------|
| Agradecimientos..... | 4 |
| Índice de figuras..... | 5 |
| Resumen y Abstract..... | 7 |
| 1. Introducción..... | 8 |
| 1.1. Objetivo..... | 8 |
| 1.2. Contexto | 8 |
| 2. Monitorización de calidad en procesos de soldadura | 10 |
| 2.1. Relevancia de los procesos de soldadura en la actualidad | 10 |
| 2.2. Evaluación de calidad..... | 11 |
| 2.3. Monitorización en tiempo real | 13 |
| 2.3.1. Sensores eléctricos..... | 14 |
| 2.3.2. Visión artificial | 15 |
| 2.3.3. Termografía infrarroja..... | 16 |
| 2.3.4. Espectroscopía óptica de plasma | 18 |
| 3. Inteligencia Artificial (IA) aplicada a la clasificación de defectos en soldadura | 20 |
| 3.1. Soluciones basadas en <i>Fuzzy Logic Control</i> | 21 |
| 3.2. Soluciones basadas en Redes Neuronales Artificiales..... | 23 |
| 3.3. Memoria Jerárquica Temporal (HTM) | 26 |
| 3.4. Deep Learning | 30 |
| 4. Implementación Software y realización de pruebas..... | 36 |
| 5. Resultados experimentales..... | 48 |
| 6. Conclusiones y lineas futuras | 52 |
| 7. Bibliografía | 53 |
| 7.1. Referencias de texto..... | 53 |
| 7.2. Referencias de imágenes..... | 55 |
| ANEXO 1 | 58 |
| ANEXO 2 | 61 |

AGRADECIMIENTOS

Quisiera comenzar agradeciendo a Jesús toda su paciencia conmigo, por ponérmelo todo tan fácil para llevar a cabo este proyecto. Gracias por ser tan cercano y amable.

También a José, con el que estuve trabajando 6 meses en el laboratorio, por tu ayuda, dedicación y por estar siempre disponible cuando lo he necesitado y me ha surgido algún problema. Hago extensible el agradecimiento a los compañeros del laboratorio.

A mis amigos de la uni, por tantos ratos en el hall, Escudero, por los desayunos y cafés donde Felipe. Gracias por los momentos de “NOTAS”, no eran situaciones de nervios por los correos del Vicerrectorado, lo eran por los vuestros en los que poníais en *Asunto*: SUSPENSO.

A los de siempre, que desde niños hemos crecido juntos, porque se que cuando necesito algo, ahí estáis. A mis amigos y hermanos internos del Colegio Torreánaz, Hermanas y profesores, gracias por inculcarme tantas cosas.

Por último a mi familia, mis padres que con tanto cariño nos han criado, espero devolveros toda la felicidad que os merecéis. A mis hermanos Manuel y Virginia, por todas las discusiones y buenos momentos, a mis abuelos que son mis segundos padres, a todos mis tíos, primos...

GRACIAS POR HACERME SENTIR TAN FELIZ

ÍNDICE DE FIGURAS

| | | |
|-----|--|----|
| 1. | <u>FIGURA 1.2.1</u> PROCESO DE SOLDADURA..... | 9 |
| 2. | <u>FIGURA 2.3.1</u> CONTROLADOR DEL LABORATORIO | 14 |
| 3. | <u>FIGURA 2.3.2.1</u> ESPECTRO DE LA LUZ..... | 15 |
| 4. | <u>FIGURA 2.3.3.1</u> DISPOSICIÓN DE LA CÁMARA IR Y PIEZA DURANTE LA INSPECCIÓN POR TERMOGRAFÍA...16 | |
| 5. | <u>FIGURA 2.3.3.2</u> EJEMPLO DE TERMOGRAFÍA PARA DETECCIÓN DE PATOLOGÍAS EN EDIFICACIÓN.....17 | |
| 6. | <u>FIGURA 2.3.4.1</u> EJEMPLO DE UN ESPECTRO DE PLASMA EMITIDO DURANTE UNA SOLDADURA.....18 | |
| 7. | <u>FIGURA 2.3.4.2</u> PASOS A SEGUIR EN EL PROCESO DE USO DE UN ESPECTRO ÓPTICO DE PLASMA | 19 |
| 8. | <u>FIGURA 3.1</u> UNIONES ENTRE NEURONAS DENTRO DEL NOCÓRTEX HUMANO..... | 21 |
| 9. | <u>FIGURA 3.1.1</u> ESTRUCTURA BÁSICA DE FUZZY LOGIC EN UN PROCESO DE SOLDADURA CON ARCO..... | 22 |
| 10. | <u>FIGURA 3.1.2</u> ESTRUCTURA DE UNA RED NEURONAL BASADA EN EL CONTROL POR FUZZY LOGIC..... | 23 |
| 11. | <u>FIGURA 3.2.1</u> ESTRUCTURA DE ACTUACIÓN DE LA NEURONA HUMANA, POR McCULLOCH Y PITTS..... | 24 |
| 12. | <u>FIGURA 3.2.2</u> REPRESENTACIÓN DE UNA RED NEURONAL A DÍA DE HOY..... | 25 |
| 13. | <u>FIGURA 3.3.1</u> DIAGRAMA DE 4 REGIONES ORGANIZADAS EN UNA JERARQUÍA DE 4 NIVELES..... | 27 |
| 14. | <u>FIGURA 3.3.2</u> REDES DE DIFERENTES SENSORES CONVERGIENDO..... | 27 |
| 15. | <u>FIGURA 3.3.3</u> ESPECTRO CAPTURADO DURANTE LA SOLDADURA QUE ALIMENTA A LOS NODOS DE LA RED HTM..... | 29 |
| 16. | <u>FIGURA 3.4.1</u> GRÁFICA QUE MUESTRA EL INCREMENTO DE LAS TÉCNICAS MACHINE Y DEEP LEARNING...30 | |
| 17. | <u>FIGURA 3.4.2</u> RED NEURONAL ARTIFICIAL CON TRES CAPAS OCULTAS | 31 |
| 18. | <u>FIGURA 3.4.3</u> ESTRUCTURA BÁSICA DE UNA CNN | 32 |
| 19. | <u>FIGURA 3.4.4</u> CAPA DE CONVOLUCIÓN-ELECCIÓN DE HIPERPARÁMETROS..... | 33 |
| 20. | <u>FIGURA 4.1</u> PROCESO A LLEVAR A CABO..... | 36 |
| 21. | <u>FIGURA 4.2</u> ESPECTRÓMETROS CONECTADOS EN FORMA DE Y..... | 37 |
| 22. | <u>FIGURA 4.3</u> LÁMPARA DE CALIBRACIÓN..... | 38 |
| 23. | <u>FIGURA 4.4</u> ELEMENTOS PARA REALIZAR LAS SOLDADURAS..... | 39 |
| 24. | <u>FIGURA 4.5</u> SOLDADURA CORRECTA..... | 40 |
| 25. | <u>FIGURA 4.6</u> SOLDADURA CON POCA INTENSIDAD DE PENETRACIÓN..... | 40 |
| 26. | <u>FIGURA 4.7</u> SOLDADURA CON MUCHA INTENSIDAD DE PENETRACIÓN..... | 40 |
| 27. | <u>FIGURA 4.8</u> SOLDADURA CON SIMULACIÓN DE SUDIEDAD..... | 41 |

| | | |
|-----|--|----|
| 28. | <u>FIGURA 4.9</u> SOLDADURAS CON CORTES DE GAS..... | 42 |
| 29. | <u>FIGURA 4.10</u> LAS LÍNEAS DE EMISIÓN DEL ARGÓN ESTÁN DISMINUÍDAS..... | 42 |
| 30. | <u>FIGURA 4.11</u> MÁQUINA QUE DOTA DE AIRE Y ARGÓN..... | 43 |
| 31. | <u>FIGURA 4.12</u> INTERFAZ DEL <i>SOFTWARE MiSpectroMon</i> | 44 |
| 32. | <u>FIGURA 4.13</u> APLICACIÓN CREADA CON <i>VISUAL STUDIO</i> | 44 |
| 33. | <u>FIGURA 4.14</u> IMAGEN 3D HECHA CON LA PALETA RAINBOW..... | 45 |
| 34. | <u>FIGURA 4.15</u> CÓDIGO C PARA CREAR LA APLICACIÓN CON VISUAL STUDIO..... | 46 |
| 35. | <u>FIGURA 4.16</u> CORDÓN DE UNA SOLDADURA | 46 |
| 36. | <u>FIGURA 4.17</u> IMAGEN PARA ENTRENAR LA RED..... | 47 |
| 37. | <u>FIGURA 5.1</u> CÓDIGO PARA REDIMENSIONAR LOAS IMÁGENES..... | 48 |
| 38. | <u>FIGURA 5.2</u> ARQUITECTURA DE LA CNN..... | 49 |
| 39. | <u>FIGURA 5.3</u> OPCIONES D EENTRENAMIENTO..... | 49 |
| 40. | <u>FIGURA 5.4</u> IMÁGENES EN LA BASE DE DATOS..... | 50 |
| 41. | <u>FIGURA 5.5</u> TABLA EN LA QUE SE MUESTRAN LOS RESULTADOS DEL ENTRENAMIENTO..... | 50 |
| 42. | <u>FIGURA 5.6</u> IMÁGENES GUARDADAS PARA EL PROCESO DE TEST..... | 51 |
| 43. | <u>FIGURA 5.7</u> PRECISIÓN TOTAL DE LA RED..... | 51 |

RESUMEN

Desde hace unos años parece incuestionable que el ámbito la de Inteligencia Artificial avanza a pasos agigantados. Las organizaciones, que están en un continuo proceso de transformación, avance y desarrollo, han sido las grandes impulsoras de esta tecnología. Esto ayuda al procesamiento de datos en unas escalas de tiempo muy pequeñas, lo que facilita la optimización y mejora de productividad de la empresa.

Por eso, actualmente se está apostando por el aprendizaje automatizado no supervisado, es decir, sin intervención humana. Gracias a Deep Learning se usa una metodología de aprendizaje parecida al que realiza el sistema nervioso humano, permitiendo que haya redes que se especialicen en la detección de determinadas características ocultas en los datos.

Un uso interesante de Deep Learning es el etiquetado de distintos objetos vistos en imágenes, el cual tiene una gran precisión. En el caso a tratar durante este trabajo, se aplicará este reconocimiento, a la detección de espectros con líneas de emisión que puedan suponer que la soldadura realizada tiene algún defecto. Esto provoca una gran reducción de tiempo e investigación de los posibles fallos que han llevado a que una soldadura no esté realizada de una manera correcta.

ABSTRACT

For some years, it seems unquestionable that the field of Artificial Intelligence is advancing by leaps and bounds. The organizations, which are in a continuous process of transformation, advancement and development, have been the great drivers of this technology. This helps the processing of data in a very small time scales, which facilitates the optimization of the data and improvement of productivity of the company.

Therefore, they are currently betting on automated learning not supervised, that is, without humane intervention. Thanks to Deep Learning, a methodology similar to that of the human nervous system is used, allowing networks to specialize in detecting certain characteristics hidden in the data.

An interesting use of Deep Learning is the labeling of different objects seen in images, which has great precision. In the case to be dealt with during this work, this recognition will be applied to the detection of spectra with emission lines that may assume that the welding is defective. This causes a great reduction of time and investigation of the possible failures that have taken to that a weld is not realized in a correct way.

1. INTRODUCCIÓN

En este primer apartado, se comentará cual es el objetivo de este trabajo, es decir, para qué puede ayudar su lectura y su posterior puesta en práctica.

También se hablará sobre el contexto en el que se ha llevado a cabo, teniendo en cuenta el tiempo en el que vivimos y lo desarrolladas que están todas las técnicas que necesitamos para realizarlo.

1.1. OBJETIVO

El objetivo de este trabajo es el diseño e implementación de una red neuronal artificial (RNA) de tipo *Deep Learning*, para su aplicación en la monitorización de la calidad en procesos de soldadura mediante espectroscopía óptica de plasmas.

Aunque ya existen estudios que han empleado enfoques similares basados en inteligencia artificial, en el presente trabajo se pretende analizar el rendimiento de una solución basada en las redes de tipo *Deep Learning*, para verificar si mejoran las prestaciones de las RNAs convencionales.

Para llegar al final y poder tratar las imágenes, se realizará una serie de pasos, desde la soldadura de varios cordones mediante un proceso de arco, hasta la captura y tratamiento de los espectros del plasma de soldadura, y su posterior conversión a imagen y todo lo que conlleva la red neuronal (diseño, entrenamiento, validación, etc.). Todo esto será comentado en los próximos apartados más detalladamente.

1.2. CONTEXTO

Para comenzar, se explicará brevemente los fundamentos y aplicaciones asociadas a espectroscopía óptica.

Al analizar las aplicaciones de la espectroscopía óptica, puede comprobarse cómo se ha empleado en muchos campos, ya sean bioquímicos, químicos, biológicos, físicos o astronómicos, debido a que un elemento puede identificarse inequívocamente a partir de sus propiedades espectrales.

En el trabajo, también se tratará el proceso de soldadura, que se puede definir como un proceso de fijación en donde se realiza la unión de dos o más piezas de un material, usualmente logrado a través de la fusión.

Muchas fuentes de energía diferentes se pueden utilizar para la soldadura, incluyendo una llama de gas, un arco eléctrico, un láser, ultrasonido, un haz de electrones, etc.

A menudo la soldadura puede realizarse en entornos muy diferentes: en el aire libre, bajo el agua y en el espacio exterior (aunque también puede hacerse en el vacío).

En la figura 1.2.1 pueden apreciarse todos los elementos que forman parte del proceso de soldadura, desde los dos materiales a soldar, el cordón mediante el que quedan unidas las dos piezas y el láser o fuente similar que provocará la fusión, pudiéndose añadir una cámara de monitorización o *tracking* [1].

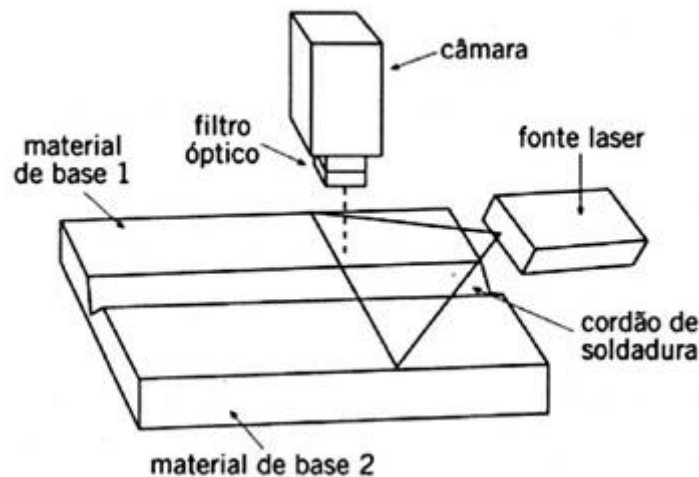


FIGURA 1.2.1 PROCESO DE SOLDADURA

En el proceso de soldadura a veces es difícil evitar la aparición de defectos debido al medio, velocidad de avance del proceso, gas de protección, corriente-tensión... Por eso, todos estos parámetros de entrada deben ser controlados.

Incluso si los parámetros de entrada están controlados, las condiciones como la temperatura, humedad, la suciedad o presencia de ciertos elementos en el ambiente, etc., pueden afectar la calidad final del proceso.

Durante este trabajo, se realizarán muchas pruebas de soldadura, dentro de las cuales, se hará tanto soldaduras correctas como cordones con diversas perturbaciones (falta/exceso de corriente, contaminación, variaciones en el flujo de gas, etc.), para así poder entrenar la red neuronal con una gran cantidad de datos y poder obtener, idealmente, un rendimiento óptimo con unos valores altos de clasificación. El objetivo final es disponer de un sistema que, de manera automática y en tiempo real, permita identificar la aparición de diferentes defectos en el proceso.

2. MONITORIZACIÓN DE CALIDAD EN PROCESOS DE SOLDADURA

2.1. Relevancia de los procesos de soldadura en la actualidad

Hay muchos sectores industriales donde la soldadura juega un papel fundamental. La soldadura se ha utilizado en la fabricación de acero, ha expandido sus usos en otros sectores industriales como la construcción, la manufactura mecánica y de coches, etc.

La soldadura comenzó a utilizarse en la Edad del Bronce y la Edad del Hierro, y desde entonces se ha ramificado en todo el mundo.

Hay tres tipos de soldadura a los que se va a hacer referencia durante este trabajo: arco, MIG y TIG. Ha habido muchos tipos diferentes de soldadura que se han consumible por objetivo a un arco eléctrico. El metal base y la adición de metales deben estar juntos cuando están cerca del electrodo. La soldadura llamada MIG (*Metal Inert Gas* / Gas Inerte y Metal) es la forma más fácil de soldar. El arco se produce mediante un electrodo formado por un hilo continuo y las piezas a unir, quedando este diversificado con el paso de los años. La soldadura por arco se realiza mediante un electrodo protegido de la atmósfera circundante por un gas inerte. La soldadura TIG (*Tungsten Inert Gas* / Gas inerte y Tungsteno) utiliza un electrodo que no se derrite hecho de tungsteno. Este tipo de proceso de soldadura es muy flexible, ya que el metal es añadido por separado.

Estos tres tipos de soldadura requieren un proveedor de energía para el mantenimiento de arco eléctrico, entre el electrodo y el material base para derretir los metales en el punto de unión. Las corrientes AC o DC pueden ser utilizadas para este tipo de soldadura.

En cada proceso se utiliza gas inerte o semi-inerte para prevenir la combustión en la zona. El soldador debe estar bien capacitado para estos tipos de trabajos. A veces, se producen muchos problemas debido a la falta de medidas de seguridad. El trabajo inseguro, es decir, sin el vidrio de soldadura y tener materiales combustibles cerca, puede ser peligroso. Al tomar las medidas preventivas, el riesgo de muerte o riesgos para la salud pueden ser reducidos.

Por ejemplo, en la soldadura por arco, el trabajador debe usar casco, guantes de mano y otros elementos de protección que ayudan al soldador a trabajar de forma segura en el proceso [2].

2.2. EVALUACIÓN DE CALIDAD

La inspección de soldaduras se puede definir como aquellas actividades que se llevan a cabo para mantener un grado de calidad y fiabilidad del proceso, para evitar errores. Esto se puede conseguir mediante la evaluación y seguimiento a lo largo de todo el proceso, solucionando aquellos errores que se pudieran cometer.

La importancia de la inspección y seguimiento radica de la responsabilidad de los equipos que actualmente se fabrican por soldadura, ya que, en determinadas condiciones de fallo, pueden afectar a la seguridad. Algunos ejemplos de estas construcciones pueden ser: aviones, buques, trenes, plantas de energía (térmicas, hidroeléctricas y nucleares), puentes, estructuras metálicas, etc. [3]

Los END (Ensayos No Destructivos) conocidos también como NDT (*Non Destructive Testing*) –por sus siglas en inglés–, son pruebas que se practican, entre otros, para detectar la aparición de defectos en las soldaduras, con el fin de conocer información relevante sobre su estado y calidad. Este tipo de ensayos, que se realizan mediante la aplicación de pruebas físicas, tales como ondas electromagnéticas, acústicas y elásticas, emisión de partículas subatómicas y capilaridad, entre otras, se consideran no destructivos, porque su práctica no representa ningún riesgo real para el objeto en estudio, ni altera de forma permanente sus propiedades físicas, químicas, mecánicas ni dimensionales.

Los END pueden aplicarse en cualquier etapa del proceso, bien sea para controlar la calidad de la materia prima, durante los procesos de producción y fabricación, como medio de control de calidad en las salidas de producción, o como método de inspección durante esquemas y servicios de mantenimiento.

Detectar a tiempo las discontinuidades, defectos e imperfecciones de los cordones de soldadura, puede marcar la diferencia entre uniones seguras, y aquellas potencialmente con riesgo o susceptibles de siniestralidad. Es por ello, que la realización de ensayos no destructivos para verificar el estado de la soldadura y de las piezas a las que pertenece, se ha convertido en una de las prácticas más utilizadas actualmente.

Los ensayos no destructivos son de gran importancia a la hora de examinar las uniones y piezas soldadas, ya que este tipo de pruebas permiten no sólo determinar la calidad y características de la soldadura, sino que además pueden alertar sobre la presencia de discontinuidades en la misma, y ofrecer datos tan relevantes como el tamaño, la forma y la situación de dichos errores.

Entre los ensayos no destructivos que se utilizan con mayor frecuencia para inspeccionar soldadura se encuentran:

- Ensayo Visual (VT): como su nombre indica, es un ensayo que se realiza mediante un proceso de inspección visual y brinda información inmediata del área evaluada. Aunque su práctica es sencilla, requiere de personal calificado, capaz de comparar los datos obtenidos en la superficie del cordón de soldadura, con los estándares nacionales e internacionales que determinan la aceptabilidad y calidad de una unión soldada.

Mediante este método se puede examinar, sobre todo, aspectos relacionados con la dimensión del cordón, su longitud y todas aquellas imperfecciones superficiales que pueden, en un futuro, causar fallos de resistencia y fatiga del material o de las uniones soldadas.

- Líquidos Penetrantes (PT): este procedimiento se utiliza básicamente para detectar discontinuidades en materiales sólidos no porosos, como el acero inoxidable, aluminio y sus aleaciones, cobre, bronce y latón; además, es uno de los métodos no destructivos más usados en la inspección de soldadura y sus acabados. Este ensayo se basa en el principio físico conocido como capilaridad, y se realiza básicamente, mediante la aplicación sobre la superficie a inspeccionar –por inmersión, brocha, pincel o pulverización–, de líquidos de baja tensión superficial que penetran los poros y son retenidos en las discontinuidades y fisuras.

Se reconoce como uno de los ensayos no destructivos más favorables, porque es económico, fácil de emplear, no requiere de equipos complejos ni costosos, permite inspeccionar la totalidad de la superficie de la pieza –sin importar su geometría ni tamaño–, y revela de manera inmediata los defectos. Sin embargo, también presenta limitaciones, ya que no puede usarse en materiales porosos, ni en superficies o piezas pintadas o con algún tipo de recubrimiento protector, pues hay riesgo de afectar al material o el recubrimiento con los líquidos.

- Partículas Magnéticas (MT): mediante este método se puede detectar las discontinuidades que se encuentran sobre o justamente debajo de la superficie. Es una prueba que se puede llevar a cabo en piezas de tamaño y forma variable, superficies con acabados soldados y en todo tipo de metales ferrosos. Para realizar esta prueba es preciso someter el cordón de soldadura a una magnetización o flujo magnético y espolvorear sobre él, partículas finas de material ferro magnético o polvo de hierro. Si en algún lugar de la superficie examinada se forma un campo de fuga que atrae hacia éste, el polvo de hierro es porque presenta una imperfección o discontinuidad.

Este ensayo es de gran utilidad a la hora de detectar imperfecciones superficiales y sub superficiales; sin embargo, tiene algunas limitantes asociadas con la dirección de las discontinuidades, ya que sólo detecta las ubicadas perpendicularmente al campo,

además sólo puede aplicarse a materiales ferro magnéticos, y tiene una capacidad de penetración limitada.

- Radiografía o Rayos X (RT): los rayos gamma, conocidos también como rayos “X”, tienen la propiedad de atravesar los materiales opacos sin reflejarse ni refractarse, para producir una impresión fotográfica. Esta cualidad de los rayos gamma permite inspeccionar internamente los cordones de soldadura y detectar defectos como grietas, bolsas e inclusiones, entre otros, ya que dichas discontinuidades absorben las radiaciones en diferente proporción del material base y generan un tipo de contrastes “claro– oscuro”, que permite identificarlas fácilmente en las radiografías.

De acuerdo con la disposición de los equipos que intervienen en la emisión de los rayos “X”, existen las siguientes técnicas de ensayo: técnica de pared simple, técnica de pared doble vista simple, técnica de exposición panorámica... [4]

Hay aparte de estos, otros muchos tipos de END, pero se ha comentado esos, como los principales y más usados.

Para estos casos es muy útil un refrán español, pero cambiando algunas palabras:

“Vale más inspeccionar que lamentar.”

Llegados a este momento, se puede apreciar lo importante y grandes aplicaciones futuras que tiene esta línea de trabajo, para detectar en tiempo real imperfecciones, errores, fallos en los procesos de soldadura realizados, sin la necesidad de realizar ensayos *a posteriori*.

2.3. MONITORIZACIÓN EN TIEMPO REAL

A lo largo de los años, la demanda de sistemas para la monitorización y el control en la soldadura ha tendido a crecer. La necesidad de realizar una soldadura de forma más rápida, eficiente y de mejor calidad, llevó a un desarrollo y perfeccionamiento de técnicas y sistemas para el control y la monitorización de ésta.

Un sistema robotizado para soldadura puede necesitar un controlador que permita ajustar los parámetros del proceso en tiempo real, cuando se detectan alteraciones relativas a los parámetros y al procedimiento predefinido. Para ejecutar estos ajustes, el sistema debe estar equipado con sensores y sistemas de control y monitorización que se comuniquen entre sí, con el equipamiento de soldadura y con el robot, de forma que permitan una respuesta en tiempo real.

En este trabajo se ha utilizado un controlador disponible en el laboratorio, el cual ayuda a la realización de las simulaciones. Cuenta con un motor para mover la antorcha a lo largo de las chapas a soldar, subirla o bajarla, etc.

Este controlador (figura 2.3.1), es manejado desde el programa *WeldStorm 2.0*, aunque se le puede añadir eventos para ser controlado de una manera automática.

Se han desarrollado muchos proyectos de investigación y desarrollo en este campo, los cuales han llevado a la aparición de una gran diversidad de tipos de sensores y controladores.



FIGURA 2.3.1 CONTROLADOR DEL LABORATORIO

2.3.1. SENSORES ELÉCTRICOS

Una vez determinados los parámetros de la soldadura es necesario asegurar que se mantendrán dentro de unos límites aceptables durante todo el proceso. Para ello, se recurre al uso de sensores que leen los parámetros referidos, dependiendo del tipo de sensor, del tipo de parámetro a medir, del control y el ajuste de los parámetros realizado por los controladores incorporados en el sistema automatizado.

En cuanto a la manera de funcionar, los sensores pueden clasificarse en sensores que entran en contacto con la pieza, de forma mecánica o eléctrica, y sensores sin contacto, por ejemplo, los ópticos e inductivos.

Para la mayor parte de los procesos de soldadura, los parámetros monitorizados y comparados con valores de referencia, son magnitudes eléctricas, como intensidad de corriente y tensión, temperatura, velocidad de soldadura, distancia entre la antorcha y la pieza, la fuerza aplicada en los electrodos y otros parámetros geométricos que caracterizan el proceso de soldadura [5].

2.3.2. VISIÓN ARTIFICIAL

Los sistemas basados en la visión del proceso en cuestión se emplean industrialmente para el control remoto del arco de soldadura. Una de las principales dificultades que se puede encontrar por parte de los operadores es la alta intensidad de luz emitida en el proceso, lo que impide una buena visibilidad de la soldadura.

La calidad de la imagen que el operador obtiene con los sistemas de vigilancia basados en la visión a menudo no es suficiente para emitir un juicio, con respecto a la necesidad de realizar ajustes o no, en los parámetros durante el proceso. Para solucionar esto, una posible solución es filtrar la mayor parte de radiación del arco, con un filtro paso banda e iluminar el área circundante con luz de diodos láser en la misma longitud de onda. La cantidad de luz capturada por la cámara depende del tiempo de obturación que tenga. La luz del láser sólo se atenúa si el ancho del pulso es mayor que el tiempo de exposición de la cámara, o si el láser es una onda continua.

El filtro, se elige en una longitud de onda donde la intensidad de luz del arco es baja. Sin embargo, dicho filtro oscurece la imagen. Para solucionar este problema, se añade una fuente de iluminación para garantizar la visión global de la soldadura. Con esto se consigue eliminar la mayoría de la contaminación lumínica que antes hacía imposible seguir el proceso.

Como consecuencia, la poca visibilidad los sistemas basados en visión, se limita en su uso industrial a una ayuda visual para el seguimiento de las articulaciones y el seguimiento cualitativo del proceso, con mejoras en los sistemas de visión, que podrían ser utilizados para el control de calidad y control durante el proceso [6].

La figura 2.3.2.1 muestra el espectro de la intensidad de la luz que hay que tener en cuenta para escoger un buen filtro y aplicar ese espectro de longitud de onda que más conviene para obtener un buen resultado.

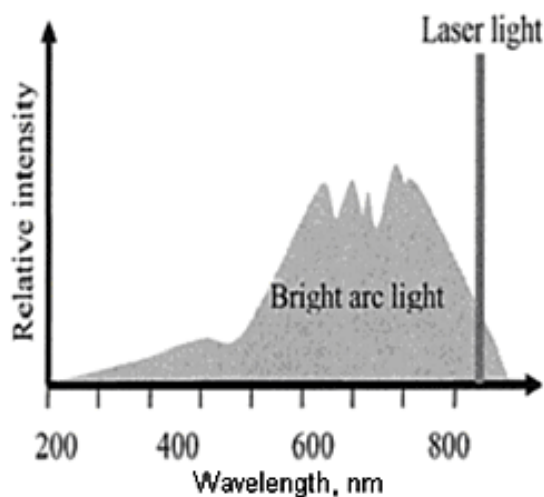


FIGURA 2.3.2.1 ESPECTRO DE LA LUZ

2.3.3. TERMOGRAFÍA INFRARROJA

La termografía infrarroja (TIR) es un método de diagnóstico en equipos, partes y componentes de un sistema, mediante la obtención de imágenes térmicas (termogramas) de la distribución de temperatura en función de la radiación infrarroja emitida por su superficie. La TIR es una técnica de ensayo no destructivo (END) sin contacto [7].

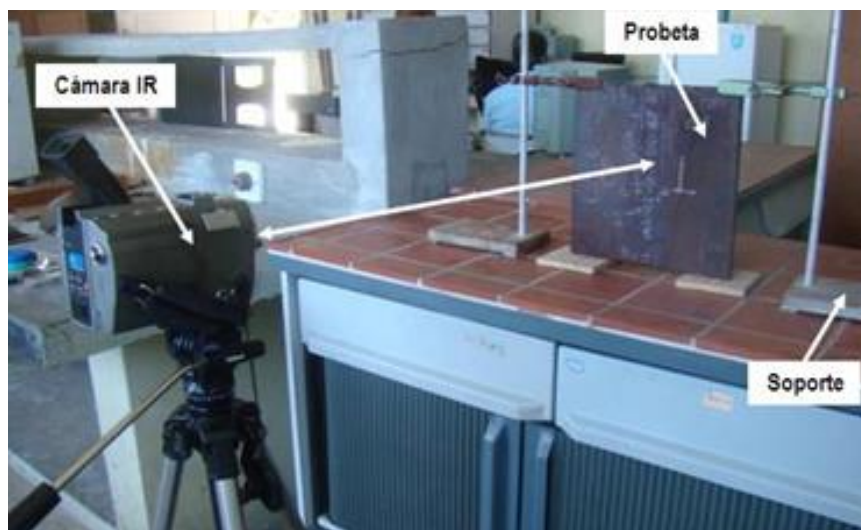


FIGURA 2.3.3.1 DISPOSICIÓN DE LA CÁMARA IR Y PIEZA DURANTE LA INSPECCIÓN POR TERMOGRAFÍA.

Cuando el flujo de calor de un material cambia por la presencia de defectos o imperfecciones en el aire o en dicho material, se puede producir contrastes de temperatura en la superficie. El uso de la TIR como método no destructivo de inspección, se basa en obtener, analizar y representar mediante las imágenes de los termogramas los distintos patrones térmicos durante el proceso.

La variación de unos pocos grados con respecto a la temperatura de referencia muestra un comportamiento no usual. Mediante la técnica TIR se puede conseguir estos valores de temperatura a una distancia prudencial del objeto, manteniéndose a salvo la persona que está realizando las medidas, y consiguiendo unos buenos resultados de la medición.

Esta técnica se puede utilizar en otros ámbitos de la vida y en otros contextos, ayudando en la detección de cáncer de pecho, informando sobre posibles desórdenes vasculares, avistando posibles enemigos en el terreno de la defensa militar, revelando humedades de edificios...

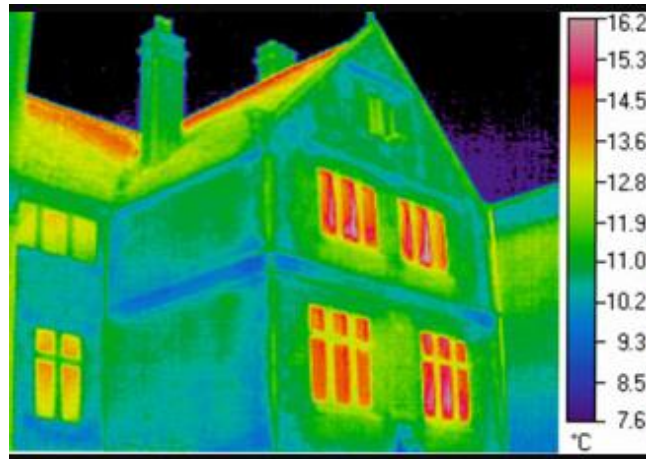


Figura 2.3.3.2 EJEMPLO DE TERMOGRAFÍA PARA DETENCIÓN DE PATOLOGÍAS EN EDIFICACIÓN

En termografía activa se usa una estimulación externa para provocar un flujo de calor interno en el objeto estudiado. Un defecto interno afectaría al flujo calorífico produciendo un contraste térmico en la superficie. Las técnicas de TIR activas principales son: TIR pulsada, *Step heating* y TIR *lock-in*.

La TIR pulsada (*Pulsed Thermography*) consiste en aplicar un pulso corto de calor sobre el objeto y grabar el enfriamiento del espécimen. El frente térmico aplicado se propaga en el material y cuando encuentra un defecto, el *ratio* de difusión es reducido, produciendo un contraste de la temperatura sobre ese punto. De esta manera, el contraste de defectos más profundos aparecerá más tarde y con menor diferencia de temperaturas.

La TIR pulsada es usada, por ejemplo, en la inspección de componentes estructurales de aviones, control de calidad de soldadura por puntos, detección de grietas o corrosión...

En la técnica de *Step Heating* o termografía de pulso largo, el objeto es calentado continuamente a baja potencia y se monitoriza el incremento de temperatura de la superficie.

Aplicaciones de esta técnica son, por ejemplo, la evaluación de espesores de recubrimientos y de uniones de recubrimiento a sustrato en estructuras compuestas y también la detección de corrosión oculta en el fuselaje de aviones.

La TIR *lock-in* está basada en la generación de ondas de calor dentro del espécimen inspeccionado y monitorizando de forma sincronizada el campo de temperaturas oscilantes, obtenido mediante una computadora o un amplificador *lock-in*. Por transformación de Fourier se obtienen las imágenes de fase y amplitud de la temperatura [8].

2.3.4. ESPECTROSCOPIA ÓPTICA DE PLASMAS

La espectroscopía de emisión óptica es en su uso común, un diagnóstico de laboratorio y de procesamiento de plasmas. Allí puede existir una gran cantidad de información en los espectros ópticos, tomados de una región relativamente estrecha del espectro. Estos espectros se pueden utilizar para numerosos procesos de control [9].

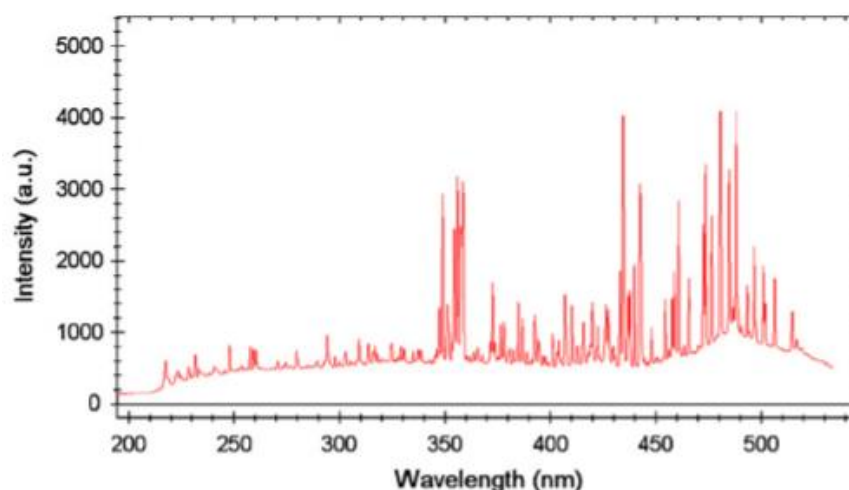


FIGURA 2.3.4.1 EJEMPLO DE UN ESPECTRO DE PLASMA EMITIDO DURANTE UN PROCESO DE SOLDADURA

El análisis de la luz emitida durante los procesos de soldadura, tales como arco o soldadura láser, también permite una monitorización de soldadura en línea (en tiempo real). Algunas propuestas iniciales fueron aquellas basadas en el empleo de fotodiodos, que permiten la integración de la radiación emitida sobre una específica longitud de onda y su conversión en una señal eléctrica, estableciendo así una correlación entre la señal de seguimiento de los productos y las perturbaciones generadas durante el proceso. Esto se reflejaría en la luz adquirida.

Estos enfoques han sido típicamente limitados al estudio de un defecto específico, por ejemplo, la falta de penetración. Soluciones basadas en fotodiodos han sido también propuestas para el control de procesos en línea.

Cuando la radiación del plasma es capturada, los espectros asociados se pueden analizar a través de la señal de sus líneas de emisión (en función de su longitud de onda) asociadas a las diferentes especies (elementos que contribuyen al plasma, es decir, los que forman la pieza de trabajo y el gas de protección empleado, en sus diferentes estados de ionización).

A pesar de que la radiación continua de plasma puede dar lugar a una correlación con la calidad de la soldadura, el procedimiento típico es llevar a cabo un análisis basado en algunas líneas de emisión, ya que su intensidad está relacionada con su participación en el plasma.

La temperatura electrónica del plasma es el parámetro espectroscópico común utilizado para el seguimiento, dada la correlación conocida entre la temperatura y la calidad de los cordones.

Se fijan distintas etapas para estimar el valor de la temperatura. Primero, hay una etapa de identificación de la línea, lo que implica la asociación de una línea de emisión con su elemento (especie) correspondiente. Este es probablemente el paso más complicado y crítico. Cabe señalar que varias líneas de emisión de diferentes componentes se pueden encontrar en rangos espectrales muy estrechos. Para hallar el tipo adecuado, se suele utilizar una base de datos muy conocida en este ámbito, como es la base de *Nist* [10], a partir de la cual se puede distinguir todos los componentes del material o materiales a tratar, comprobando la relación entre las intensidades de unas líneas con otras.

Vale la pena señalar que una de las principales ventajas de este enfoque radica en la reducción de costes de la configuración requerida, que puede estar formada por alguna óptica de entrada para recoger la radiación del plasma, una fibra óptica para guiar la luz desde el entorno del plasma a un espectrómetro CCD, que será controlado por algún hardware específico. El espectrómetro es típicamente el dispositivo más caro de este *setup*, y su coste estará directamente relacionado con su resolución espectral. En este trabajo se usarán dos espectrómetros conectados en Y para recoger varias medidas de una misma soldadura, y poder utilizar aquella que en cada caso sea más útil, dependiendo de la intensidad de las líneas de emisión y rango en el que trabaje cada espectrómetro.

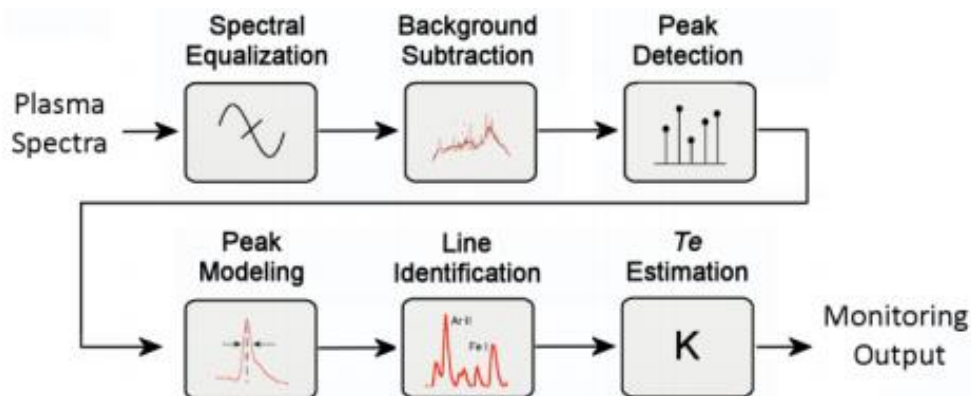


FIGURA 2.3.4.2 PASOS A SEGUIR EN EL PROCESO DE USO DE UN ESPECTRO ÓPTICO DE PLASMA

También se podría dar otros enfoques, a parte del de la identificación de las líneas y pasos sucesivos, por ejemplo, podemos tener en cuenta la intensidad de la línea de emisión y la intensidad asociada a la radiación de fondo para estimar la temperatura del plasma.

Este enfoque podría variar la búsqueda de una sensibilidad mejorada, y puede ser especialmente útil cuando se estudian nuevos procesos, donde no hay conocimiento a priori de los elementos de líneas de emisión pertinentes para ser analizados [11].

3. INTELIGENCIA ARTIFICIAL (IA) APLICADA A LA DETECCIÓN DE DEFECTOS DE SOLDADURA

Cuando la temperatura electrónica del plasma o un parámetro espectroscópico similar se emplean para el diagnóstico de la soldadura en línea, por lo general se requiere una etapa de procesamiento para indicar automáticamente si un defecto ha aparecido en el proceso. Esto es llevado a cabo normalmente a través de un análisis estadístico de las señales espectroscópicas, pero es complicado realizar la clasificación de los defectos con este tipo de enfoques.

Sin embargo, la Inteligencia Artificial (IA) usa como soluciones *Fuzzy Logic Control*, Redes Neuronales Artificiales (RNAs) o Memorias Jerárquicas Temporales (HTMs), que pueden permitir la clasificación de defectos deseada.

La lógica difusa o *Fuzzy Logic* es una solución original que se puede usar para el seguimiento y control de la soldadura. La técnica RNA también ha atraído mucha atención, debido a su capacidad inherente para mapear los problemas no lineales y la posibilidad de permitir una clasificación de salida, entre otras características.

Un problema que hay que resolver cuando una RNA va a ser utilizada para procesar espectros de plasma, es la alta dimensionalidad de los datos de entrada, que normalmente es un problema para estas redes. Técnicas de compresión como PCA (Principal Component Analysis / Análisis de Componentes Principales) se pueden utilizar en un paso anterior para reducir la dimensionalidad de los datos disponibles, mediante la eliminación de datos redundantes sin perder información importante.

Otras alternativas importantes a reseñar son los algoritmos de selección de características como SFFS (Selección Flotante de la Siguiente Secuencia), que se han utilizado también en diseños similares. Este algoritmo es de especial interés en este marco, ya que tiene en cuenta una lista de longitudes de onda en términos de su capacidad de discriminación entre las clases definidas (soldadura correcta y diferentes defectos).

La solución llamada Memoria Jerárquica Temporal (HTM) es un reciente modelo computacional propuesto por *Hawkins* y *Blakeslee*, que se inspira en la estructura del neocórtex humano, y que también ha sido empleado en este contexto.



FIGURA 3.1 UNIONES ENTRE NEURONAS DENTRO DEL NOCÓRTEX HUMANO

Una vez hecho este breve resumen, se hará más enfoque en las principales de ellas que han tenido más sentido en el aprendizaje de esta materia, para la posterior realización de este trabajo de investigación.

3.1. SOLUCIONES BASADAS EN FUZZY LOGIC

La Lógica Difusa (FL en inglés) se está convirtiendo en una técnica atractiva para el control de procesos en la soldadura, principalmente debido a su capacidad para resolver problemas en la ausencia de un modelo matemático preciso, ya que el proceso de soldadura es complejo, porque cuenta con una variable invariante en el tiempo, no lineal y la interferencia de otros muchos factores.

Con el rápido desarrollo de la producción moderna, la calidad de la soldadura se ha convertido cada vez en algo más exigente para asegurar la estabilidad del proceso de soldadura.

La lógica difusa es un enfoque interesante para un proceso como el que se está estudiando. *Langari* (1990) y *Satoshi Yamame* (1993) han mostrado resultados alentadores con el control de baño de soldadura por medio de FLC (*Fuzzy Logic Control*).

En la estrategia de control de lógica difusa, las funciones de pertenencia de los conjuntos difusos son predeterminadas por el experto antes del proceso de control en línea. Puesto que estas funciones de pertenencia normalmente permanecen sin cambios durante el proceso de control, puede conducir a resultados insatisfactorios cuando la planta controlada es invariable en el tiempo o cuando se somete a perturbaciones.

Además, la determinación de la regla difusa y la función de pertenencia, dependen en gran medida del juicio de los expertos. Los resultados satisfactorios no pueden ser posibles si el nivel de conocimientos no es el adecuado. Por eso, existe una necesidad de sintonización automática en línea de las funciones de pertenencia que se basan en diferentes características de la planta y los requisitos de rendimiento predefinido.

También sería beneficioso desarrollar un método formal para transformar el conocimiento o la experiencia humana en base a reglas de FLC para tal aplicación. Esto requiere que debe haber una disposición para detectar los cambios en las funciones de pertenencia y para determinar automáticamente la base de reglas de la FLC, utilizando los datos obtenidos directamente del sistema de soldadura por arco.

Para entender mejor como funciona este proceso, a continuación se va a explicar un ejemplo, en el cual hay que diagnosticar el grado de afectación de un material. Esto lo va a realizar un experto en base a su experiencia y conocimientos.

Hay que tener en cuenta que se pueden añadir errores por subjetividad e incertidumbre a la hora de emitir un veredicto, y gracias a *Fuzzy Logic* va a ser más fácil hacerlo.

La red neuronal artificial reconoce y clasifica patrones de comportamiento en imágenes en base al material. El sistema de lógica difusa es usado para estimar el grado de afectación del material tomando como valores el patrón de comportamiento y la cantidad de elementos.

Por tanto, gracias a la lógica difusa se dispone de sistemas expertos, que son programas computacionales que tienen disponibles las habilidades de los expertos. Estos tratan de razonar y emular los patrones que seguiría la persona encargada del trabajo, siguiendo reglas, redes asociativas, redes neuronales...

Tras todo esto, en el ejemplo anteriormente comentado, se produce una fase de validación, comparando los resultados con los de un experto en la materia [12]

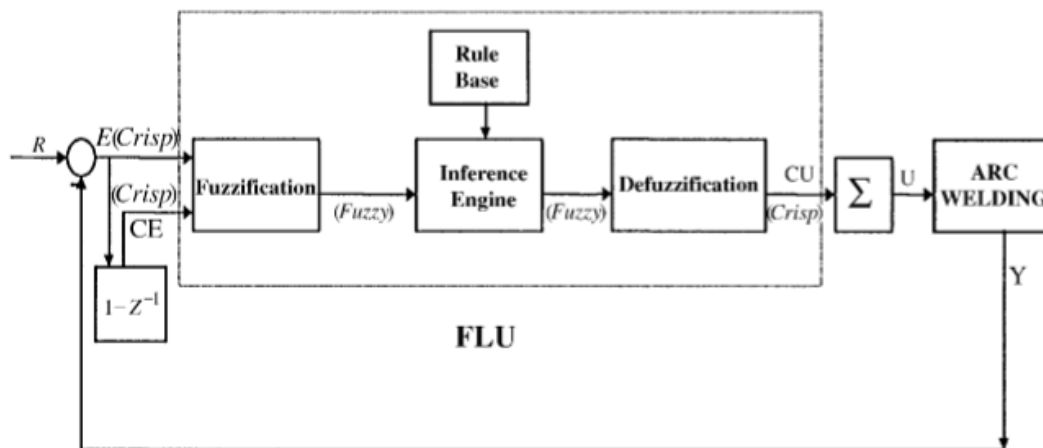


FIGURA 3.1.1 ESTRUCTURA BÁSICA DE FUZZY LOGIC EN UN PROCESO DE SOLDADURA CON ARCO

En la figura 3.3.1 se puede ver un esquema de la estructura básica de control por Fuzzy Logic de un sistema de soldadura por arco. Como lo principal de este proceso es el control, se escoge como entradas el error (E) y el cambio de error (CE) y como salida el cambio de entrada de control (CEC). Como entradas de control para el proceso, se puede tener en cuenta la velocidad, corriente de soldadura...

A continuación se adjunta la imagen 3.1.2 de una red neuronal basada en Fuzzy Logic, y una explicación de cómo se define.

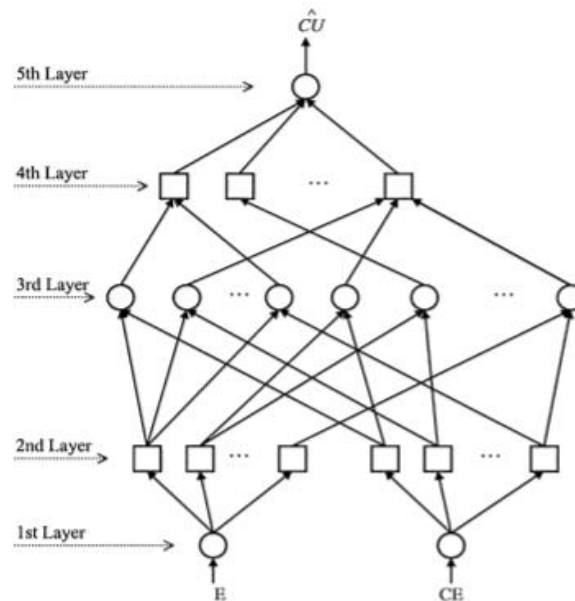


FIGURA 3.1.2 ESTRUCTURA DE UNA RED NEURONAL BASADA EN EL CONTROL POR FUZZY LOGIC

Como se puede apreciar, el ejemplo anterior consta de cinco capas , y cada una de ellas lleva las funciones de la FLC.

Los nodos se muestran como cuadrados o círculos para poder distinguirlos con más facilidad. Los cuadrados son nodos que requieren una regulación, los círculos, por contrario, aquellos que permanecen invariantes durante todo el proceso. Las flechas de las líneas que unen las capas, indican el sentido del flujo de la señal. Más adelante se incidirá más a fondo en esta cuestión [13].

3.2. SOLUCIONES BASADAS REDES NEURONALES ARTIFICIALES (RNAs)

Las RNAs son parte de otra línea de investigación de la Inteligencia Artificial (IA), que tiene como fin la construcción de máquinas inteligentes para realizar diferentes procesos, como la clasificación de patrones en diferentes clases o categorías. Los orígenes de la IA se pueden encontrar en la década de los 40, ya que ahí se originó un movimiento científico llamado ***cibernética***, el cual se creó en torno a que la idea del funcionamiento de sistemas vivos o artificiales puede ser mejor si tenemos más en cuenta los modelos basados en la transferencia de información, que aquellos basados en la transferencia de energía.

Por eso se comenzó con el estudio del sistema nervioso humano. A esta investigación contribuyeron matemáticos, biólogos, filósofos...

En los 50, surgieron dos perspectivas, la IA basada en el procesamiento simbólico y las redes neuronales. Nosotros nos centraremos especialmente en esta última.

Un paso importante, fue entender cómo funcionaban las neuronas humanas, para ello *McCulloch y Pitts*, presentaron una estructura que se puede ver en la figura 3.2.1 de a continuación, donde si la suma de las entradas excitatorias supera el umbral de activación de la unidad y además no hay entrada inhibitoria, se activará la neurona y emitirá respuesta. Por el contrario, no se activará. Combinando varias neuronas, se dará lugar a una función lógica finita.

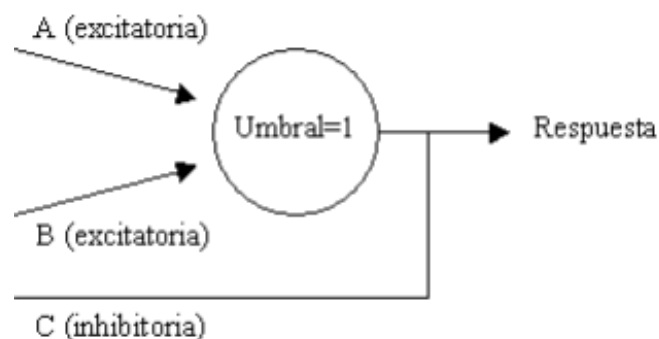


FIGURA 3.2.1 ESTRUCTURA DE ACTUACIÓN DE LA NEURONA HUMANA, POR McCULLOCH Y PITTS

A partir de ahí continuó el proceso de investigación, y poco a poco se llegó al planteamiento de las redes neuronales y su implementación, como ayuda a la realización de trabajos en distintos campos y sus aportaciones a mejorar la simplicidad y optimización en dichos procesos [14].

Como se ha comentado anteriormente, una red neuronal es una herramienta muy útil, porque los parámetros de entrada y salida pueden relacionarse y compararse con el valor que viene dado por la red neuronal. El fin último de la RNA, es hacer un uso optimizado de los parámetros. Este proceso está inspirado en las redes neuronales biológicas.

En la mayoría de los casos, una red neuronal es un sistema que se adapta y cambia su estructura durante la fase de aprendizaje. Dicho aprendizaje, es una parte muy importante del proceso.

A veces, estos sistemas son también asociados con un algoritmo de aprendizaje. Un uso común de la RNA es la definición de una clase con una serie de funciones, en las que los miembros de la clase se obtienen mediante la variación de los parámetros, la conexión de los pesos o detalles específicos de la arquitectura, tales como el número de neuronas o su conectividad.

Una representación de una RNA se puede ver en la imagen 3.2.2.

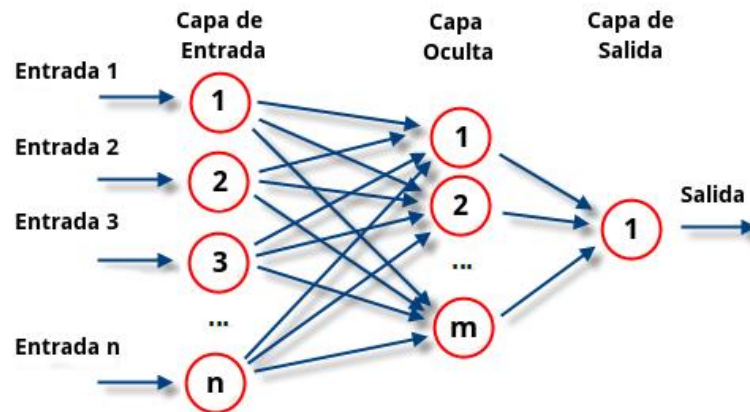


FIGURA 3.2.2 REPRESENTACIÓN DE UNA RED NEURONAL A DÍA DE HOY

En esta red neuronal se puede distinguir tres tipos de capas:

- El primer tipo es la **capa de entrada**, que está formada por aquellas neuronas que introducen los valores de entrada en la red. En estas neuronas no se produce procesamiento.
- La siguiente zona para explicar, es la **capa oculta**, que está formada por aquellas cuya entrada es la salida de las capas de entrada y cuyas salidas pasan a la capa de salida.
- Por último, la **capa de salida**, formada por neuronas cuyos valores de salida corresponden a las salidas de la RNA.

En este tipo de redes se suele usar la *propagación hacia atrás*. Una vez que se ha aplicado un patrón a la entrada de la red como estímulo, este se propaga desde la primera capa a través de las capas superiores de la red, hasta generar una salida. La señal de salida se compara con la salida deseada y se calcula una señal de error para cada una de las salidas.

Las salidas de error se propagan hacia atrás, partiendo desde la capa de salida hacia todas las neuronas de la capa oculta que contribuyen directamente a la salida. Sin embargo, las neuronas de la capa oculta sólo reciben una fracción de la señal total del error, basándose aproximadamente en la contribución relativa que haya aportado cada neurona a la salida original. Este proceso se repite, capa por capa, hasta que todas las neuronas de la red hayan recibido una señal de error que describa su contribución relativa al error total.

La importancia de este proceso consiste en que, a medida que se entrena la red, las neuronas de las capas intermedias se organizan a sí mismas, de tal modo que las distintas neuronas aprenden a reconocer distintas características del espacio total de entrada.

Después del entrenamiento, cuando se les presente un patrón arbitrario de entrada que contenga ruido o que esté incompleto, las neuronas de la capa oculta de la red responderán con una salida activa si la nueva entrada contiene un patrón que se asemeje a aquella característica, el cual las neuronas individuales hayan aprendido a reconocer durante su entrenamiento [15].

A continuación, se explicará dos importantes análisis a utilizar dentro de las Redes Neuronales Artificiales.

3.3. MEMORIA JERÁRQUICA TEMPORAL (HTM)

La Memoria Jerárquica Temporal (HTM) es una tecnología de aprendizaje de máquina, que pretende capturar las propiedades estructurales y algorítmicas del neocórtex. Las HTMs pueden verse como un tipo de red neuronal. Por definición, cualquier sistema que trate de modelar los detalles arquitecturales del neocórtex, es una red neuronal.

Las HTM modelan neuronas (llamadas células en terminología HTM), las cuales son organizadas en columnas, capas, regiones y en una jerarquía. Los detalles son importantes, en este respecto, las HTMs son una nueva forma de red neuronal.

Las HTMs son entrenadas con muchos tipos de datos variados y confían en el almacenamiento de grandes grupos de patrones y secuencias. El modo en el que los datos son guardados y accedidos es lógicamente diferente del modelo estándar utilizado por los programadores. Un programador puede implementar cualquier tipo de organización y estructura de datos ya que tienen control sobre cómo y dónde es guardada la información.

Por contraste, la memoria HTM es más restrictiva, ya que tiene una organización jerárquica y está basada en el concepto de tiempo. El usuario especifica el tamaño de la jerarquía y en qué entrenar al sistema, pero la HTM controla dónde y cómo se almacena la información.

Una red HTM consiste en regiones organizadas en una jerarquía. La región es la unidad principal de memoria y predicción en una HTM.

Típicamente, cada región HTM representa un nivel en la jerarquía. A medida que se va ascendiendo, existe convergencia, esto es, múltiples elementos en una región hijo convergen en un elemento de una región padre. De todas maneras, debido a las conexiones de retroalimentación, la información también diverge a medida que se desciende en la jerarquía.

Se usa la palabra “región” al describir la función interna, y usamos la palabra “nivel” cuando nos referimos específicamente al rol de la región dentro de la jerarquía.

En el ejemplo de la imagen 3.3.1, se puede ver una representación de una jerarquía de cuatro niveles, comunicando información dentro de cada nivel, entre niveles y desde/hacia el exterior de la jerarquía.

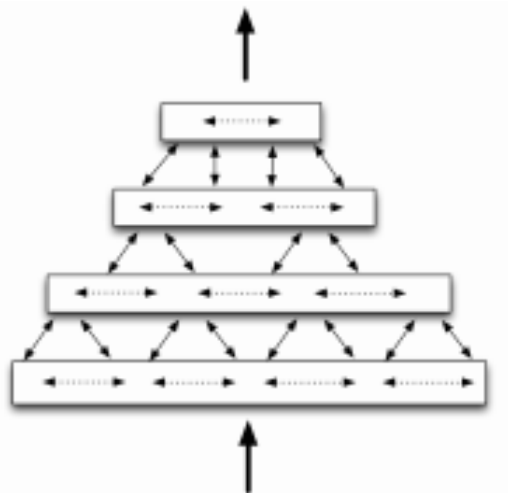


FIGURA 3.3.1 DIAGRAMA DE 4 REGIONES ORGANIZADAS EN UNA JERARQUÍA DE 4 NIVELES

Es posible combinar múltiples redes HTM. Este tipo de estructura tiene sentido si se tienen datos de más de una fuente o sensor.

Por ejemplo, una red podría estar procesando información auditiva y la otra red información visual. Existe convergencia dentro de cada red, pero en las redes separadas se juntan únicamente en la parte de arriba. Esto se puede apreciar en la imagen 3.3.2.

El beneficio de la organización jerárquica es la eficiencia. Reduce significativamente el tiempo de entrenamiento y el uso de la memoria, ya que los patrones aprendidos en cada nivel de la jerarquía son reutilizados cuando se combinan de manera distinta en los niveles superiores.

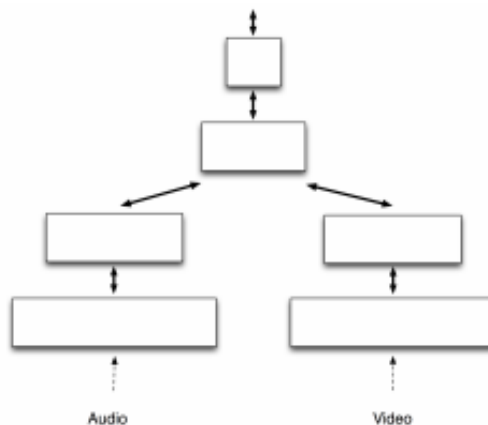


FIGURA 3.3.2 REDES DE DIFERENTES SENSORES CONVERGIENDO

Las HTM aprenden automáticamente las mejores posibles representaciones en cada nivel, conociendo las estadísticas de los datos de entrada y el número de recursos reservados. Si se reserva más memoria a un nivel, ese nivel formará representaciones mayores y más complejas, lo cual indica que podrían hacer falta menos niveles jerárquicos. Si se reserva menos memoria, el nivel formará representaciones más pequeñas y simples, lo que significa que podrían ser necesarios más niveles jerárquicos.

En resumen, las jerarquías reducen el tiempo de entrenamiento, reducen el uso de memoria, e introducen una nueva forma de generalización. No obstante, cabe destacar que muchos problemas simples, que conllevan predicción, pueden ser resueltos con una única región HTM.

A continuación, se comentará un poco cómo funciona y se realiza el aprendizaje de las redes HTM.

Una región HTM aprende sobre su mundo encontrando patrones y luego secuencias de patrones en la información sensorial. La región no sabe qué representa la información de entrada, ya que trabaja en un campo puramente estadístico. Busca combinaciones de entrada que ocurren juntas normalmente, lo que se llama patrones espaciales. Entonces busca en qué secuencia aparecen estos patrones en el tiempo, lo que se llama patrones temporales o secuencias.

Una simple región HTM tiene una capacidad de aprendizaje limitada. Una región, ajusta automáticamente lo que aprende, basándose en cuánta memoria tiene y la complejidad de la entrada que está recibiendo. Los patrones espaciales aprendidos por una región, necesariamente se simplificarán si la memoria asignada a una región ha sido reducida.

Los patrones espaciales aprendidos, se complicarán si la memoria asignada es incrementada. Igual que un sistema biológico, los algoritmos de aprendizaje en una región HTM son capaces de aprender en tiempo real.

A medida que los patrones de la entrada vayan cambiando, la región HTM cambiará gradualmente también. Después del entrenamiento inicial, una red HTM puede continuar aprendiendo o puede ser desactivado. Otra opción es desactivar el aprendizaje sólo en los niveles más bajos de la jerarquía, pero continuar aprendiendo en los niveles superiores.

Una vez que una HTM ha aprendido la estructura estadística básica, casi todo el aprendizaje nuevo ocurre en los niveles superiores de la jerarquía. Si un HTM es expuesto a nuevos patrones que contengan nuevas estructuras en los niveles bajos, le llevará más tiempo al HTM aprender estos nuevos patrones [16].

Se puede hacer hincapié en un ejemplo de una red HTM aplicada al diagnóstico de soldaduras. Dicho ejemplo se basa en analizar el rendimiento debido a perturbaciones provocadas en el proceso, el cual es parecido al que se sigue en este trabajo, ya que trata de diferenciar los distintos defectos que se pueden producir al soldar.

La red HTM es alimentada con estos datos (se puede ver en la figura 3.3.3 cómo el espectro alimenta a los nodos inferiores, y los superiores ofrecen el diagnóstico de la red).

La estrategia elegida es aplicar los la datos espectrales disponibles para las entradas HTM al mapear cada pixel espectral a una entrada de los nodos inferiores, ya que estos nodos tienen una gran capacidad para manejar grandes vectores a entrada.

Algunos parámetros fueron cambiados para generar los distintos defectos, tales como modificar la corriente de soldadura, perturbaciones del flujo de gas de protección o desviaciones en la trayectoria de la antorcha sobre la junta. Para cada tipo se asignó un numero de categoría [0...6].

Los resultados obtenidos no son consecuencia de un algoritmo optimizado, sino que son sólo el resultado de un estudio de viabilidad de la red HTM aplicado a la espectroscopía de plasma, donde existe una influencia producida por diferentes parámetros en la precisión de la clasificación final [17].

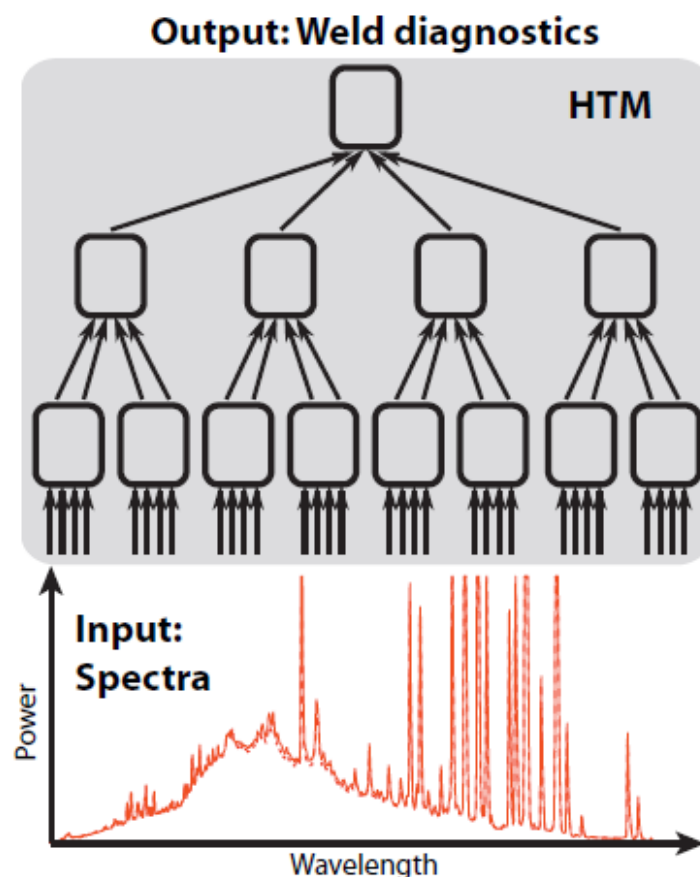


FIGURA 3.3.3 ESPECTRO CAPTURADO DURANTE LA SOLDADURA QUE ALIMENTA A LOS NODOS DE LA RED HTM

"Feasibility study of Hierarchical Temporal Memories applied to welding diagnostics" Luis Rodríguez-Cobo, Rubén Ruiz-Lombera, Olga M. Conde, José-Miguel López-Higuera, Adolfo Cobo, Jesús Mirapeix*

3.4. DEEP LEARNING O APRENDIZAJE PROFUNDO

Para finalizar el tema de la inteligencia artificial y redes neuronales, se tratará el aspecto en que está orientado este trabajo, explicando qué es, cómo funciona y cómo se aplica a la soldadura para detectar defectos.

En nuestros días, *Deep Learning* es un término que se puede decir que está de moda. Las grandes marcas y empresas mundiales del sector apuestan por este método para desarrollar funciones, aplicaciones, etc.

El aprendizaje profundo no es una idea de trabajo novedosa. La idea surgió alrededor de los años ochenta de la mano del investigador japonés *Kunihiko Fukushima*, que propuso un modelo neuronal de entre cinco y seis capas al que denominó *neocognitrón*.

Sin embargo, las dificultades para el desarrollo de alternativas a la propuesta de *Fukushima* han sido muy complejas y el coste para su investigación era sumamente elevado, lo que ha provocado que estas técnicas no se hayan vuelto a retomar con fuerza hasta hace escasamente una década, donde se ha reactivado el interés y la inversión por parte de las empresas.

Google desarrolló con éxito redes neuronales que reconocen voces en teléfonos *Android* e imágenes en *Google Plus*. *Facebook* usa *Deep Learning* para orientar los anuncios e identificar rostros y objetos en imágenes y videos. También otros como *Microsoft* o *Baidu*, lo usan.

Deep Learning es una técnica de *Machine Learning* basada en redes neuronales. Como se puede apreciar en la imagen 3.4.1, hay un aumento del uso de este tipo de algoritmos conforme pasan los años.

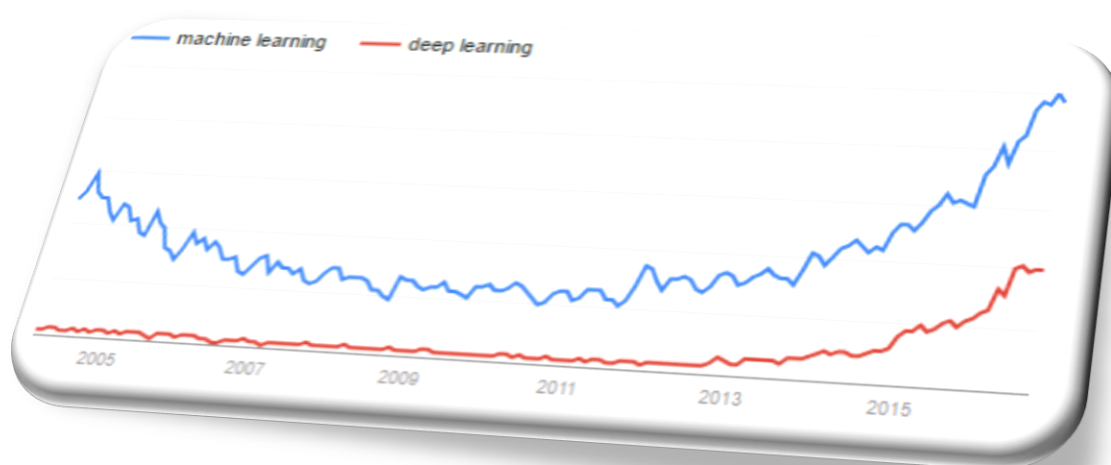


FIGURA 3.4.1 GRÁFICA QUE MUESTRA EL INCREMENTO DE LAS TÉCNICAS MACHINE Y DEEP LEARNING

Deep Learning trata de simular la forma que tiene de aprender el cerebro. Ante un estímulo, ciertas neuronas se activan, evalúan la información que han recibido, reaccionan y se comunican con otras neuronas. En posteriores estímulos añaden nuevos datos a los que ya conocen, evalúan el resultado de las acciones anteriores y corrigen su funcionamiento para tener la mejor reacción posible. Se puede decir, que es un conjunto de algoritmos de aprendizaje automático, que llevan tareas de práctica de principio a fin.

Las redes neuronales artificiales actúan de forma similar pero simplifican la maraña de neuronas en capas con diferentes objetivos. Cada grupo de neuronas analiza los estímulos de entrada (datos), los procesa y los entrega a otro grupo de neuronas en forma de datos. Este proceso se repite hasta que se activan las neuronas de salida.

Como se aprecia en el ejemplo 3.4.2 de la siguiente imagen, hay una capa que hace la función de entrada, a la vez que hay otra que hace de salida. En el medio se sitúan tres capas ocultas que relacionan la información tras el entrenamiento de la red neuronal. La cantidad de conexiones que hay entre neuronas hacen posible un buen resultado de esta técnica.

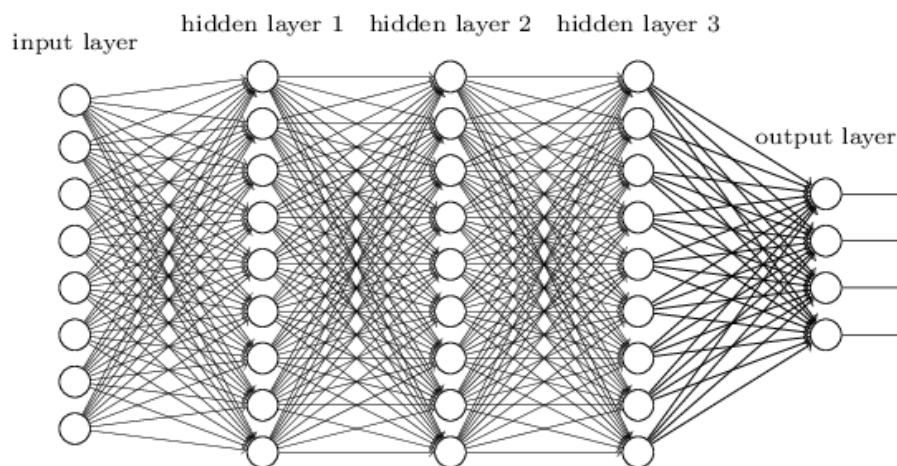


FIGURA 3.4.2 RED NEURONAL ARTIFICIAL CON TRES CAPAS OCULTAS

Deep Learning funciona gracias a la arquitectura de la red y la rutina de optimización aplicado a dicha arquitectura. Como la red es un grafo dirigido, cada capa oculta está conectada a otras muchas por debajo de ella, lo que facilita que cada capa oculta pueda ir más lejos en la red, debido a todas las combinaciones y recombinaciones de las salidas con las capas posteriores, y gracias también a sus funciones de activación.

El gran avance del *Deep Learning* respecto a otras técnicas de *Machine Learning*, se basa en que el sistema se puede entrenar, o incluso puede entrenarse a sí mismo para encontrar coherencia en datos aleatorios, de la misma forma que un bebé aprende a organizar los datos que llegan a sus sentidos.

Viendo las grandes expectativas de crecimiento de este tipo de soluciones, eminentes científicos y tecnólogos como *Stephen Hawking* o *Bill Gates* augurar un futuro catastrófico. “Las formas primitivas de Inteligencia Artificial de las que disponemos están siendo muy útiles, pero creo que el desarrollo completo de la Inteligencia Artificial podría suponer el fin de la raza humana”, aseguraba *Hawking* en una entrevista. *Hawking* basa su argumentación en que una vez se haya desarrollado la Inteligencia Artificial, los robots se rediseñarán autónomamente cada vez a mayor velocidad. Los humanos, limitados por la lenta evolución biológica, no podrán competir con ellos y serán eliminados [18].

En lo que a diseño y software se refiere, existe una fuerte comunidad sobre *Deep Learning* dentro del mundo del software libre, para potenciar e implementar soluciones y proyectos a estos niveles.

Lenguajes de desarrollo como *C++* o *Python* se nutren cada vez más de librerías fabricadas por la comunidad para lograr avances significativos en *Machine Learning*, lo que se traduce en grandes soluciones que abarcan el procesamiento del lenguaje, el reconocimiento de voz, la visión por computador, la minería de datos o el internet de las cosas.

Una vez hecho este breve resumen, se hablará más de las redes a usar en el reconocimiento y entrenamiento, para llegar a una clasificación y posterior conclusión sobre el tema a tratar.

La Red Neuronal Convolutiva es aquella que aprende por sí sola. Se llama *Convolutiva* porque puede trabajar con datos estructurados como imágenes, con el fin de clasificarlas. La diferencia con otros modelos de redes neuronales tradicionales es que aquí hay un número de capas elevado, entre 5 y 50 capas o incluso más, por eso es necesario tener en cuenta el papel que va a desempeñar la GPU (Unidad de Procesado Gráfico) en el proceso, ya que puede ser muy pesado.

La siguiente imagen muestra cómo puede ser la estructura básica de una CNN (*Convolutional Neural Network*), en la que se ve cómo viajan los datos a través de ella.

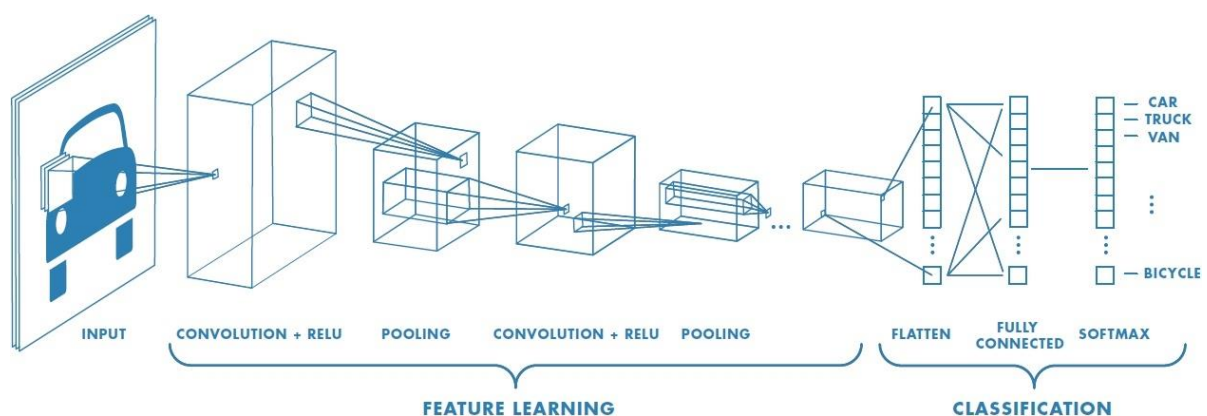


FIGURA 3.4.3 ESTRUCTURA BÁSICA DE UNA CNN

También se puede observar que las capas de más alto nivel llevan a cabo el estudio de las características del objeto, y las capas más bajas, clasifican y deciden qué es esa imagen. Para llevar a cabo un buen proceso de entrenamiento, se debe ajustar bien los pesos de todas las capas, ya que se entrenarán a la vez.

Las capas altas están formadas por una secuencia de capas con función de convolución *ReLU* (Capas de Unidad Lineal Rectificada) que no añaden complejidad a la red y *Pooling*, estas últimas reducen la dimensión de los datos y número de parámetros, calculando el máximo de una región, la media... Lo que se hace realmente es concatenar grupos de estas capas, las cuales son las encargadas del aprendizaje de características de las imágenes.

Al final de esta fase, hay un volumen, que será aplanado y convertido en un vector por la capa llamada *Flatten*, con lo que se consigue poder trabajar con las redes neurales, las cuales están totalmente conectadas. La última capa *Softmax*, indicará la etiqueta a la que corresponde la imagen, aplicando una probabilidad de semejanza. Por tanto, se puede decir que estas últimas capas llevan a cabo la clasificación.

A la CNN se le puede añadir una serie de filtros que serán aprendidos en la fase de entrenamiento, para desechar valores que no sirvan de los datos estudiados. El número de filtros es un hiper-parámetro, que indica cual es el volumen de capacidad de la capa de salida.

Otros hiper-parámetros son: el tamaño del filtro, el intervalo de repetición del filtro y los ceros de relleno para conservar el tamaño del valor de entrada o para evitar una reducción temprana del tamaño de la red, lo que llevaría a obtener peores resultados. Todo esto se resume en la siguiente imagen.

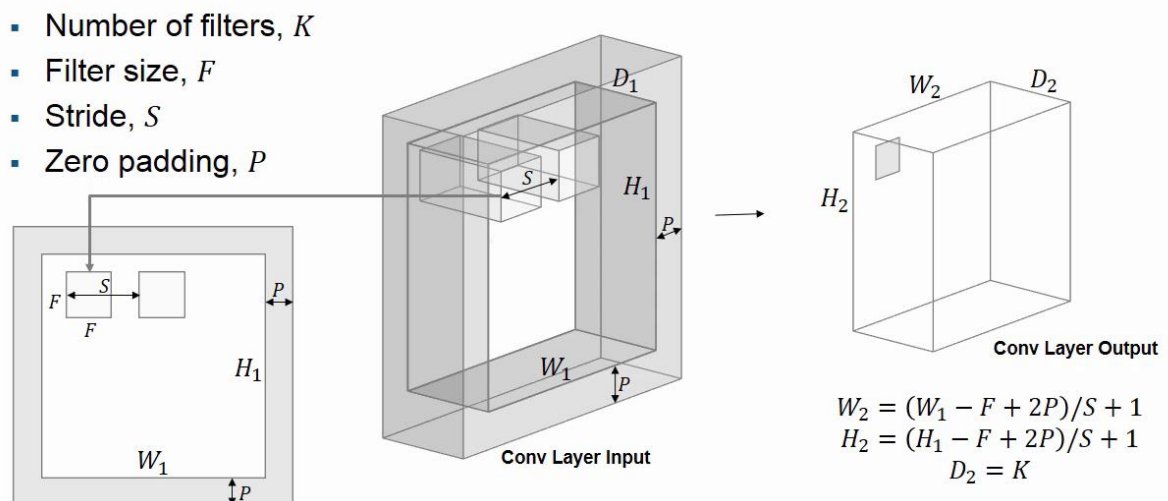


FIGURA 3.4.4 CAPA DE CONVOLUCIÓN-ELECCION DE HIPER-PARÁMETROS

La capa de convolución es el bloque principal de una CNN. El modo en que funciona es que se lleva a cabo una convolución a partir de un filtro definido, con el volumen de datos que se encuentra en la parte central de la imagen anterior, para obtener el volumen de datos de la parte derecha de la imagen. Esto se produce deslizando el filtro, el modo se puede indicar como parámetro, por cada una de las porciones que interesan del volumen de entrada, después calcular el producto escalar de la región que se está cubriendo con los valores del filtro y después hacer la suma. Esto se hará con todos los filtros que se tengan en cada caso [19].

Algunos investigadores, se han dado cuenta que usando *Deep Learning*, se llegan a soluciones más rápidas que usando otras técnicas de visión artificial. Ahora mismo, *Deep Learning* tiene una tasa de erros de menos del 5 %, mientras que otras técnicas de visión artificial se encuentran sobre el 20 % de error.

Para reducir estos errores, se introducen mayores cantidades de datos para entrenar la red neuronal, para que se pueda comparar con más valores dentro de esa base de datos y así, minimizar su *error rate*.

Dentro de DL, se puede tener dos posibles enfoques a utilizar:

- **Entrenar una red neuronal desde cero:**

En esta situación se usan desde miles a millones de datos para entrenar la red, lo que conlleva a la necesidad de días o incluso semanas para llevar a cabo problemas reales. Se consigue una gran exactitud en el proceso con este método.

- **Modificar una red ya entrenada para otros fines:**

Usando una red ya entrenada, y que tiene otras características, se cambian los pesos de las últimas capas de la red, para adecuarlos a nuevos datos. Se supone la nueva red, más pequeña y con menos datos que la primera. Se puede apreciar que la precisión de este modelo va a ser directamente proporcional a la calidad de la red pre-entrenada. Esta técnica se conoce en inglés como *Transfer Learning*.

Para llevar todo esto a cabo, se usa *Matlab* como compañero ideal en cualquier desarrollo matemático, ya que tiene una serie de funciones y *toolbox* (puede ser muy útil *Neural Network Toolbox*), que hacen el trabajo un poco más fácil. También hay *toolbox* específicas para *Deep Learning*, como puede ser *Alexnet*.

En el caso a tratar se usará un código con una serie de parámetros configurados para asegurar una mejor solución y un menor error de clasificación. Este código se recoge en el **Anexo 1** al final del trabajo.

A la hora del entrenamiento, el cual podría llevar días si existe una base de datos grande y una red neuronal con varias capas, es clave la automatización y puede ayudar el descargar los cálculos en un clúster, donde haya varias GPUs disponibles, ya que permitirá entrenar varias arquitecturas en paralelo, de modo que se acorta el tiempo que lleva encontrar la arquitectura idónea para el problema.

Se puede sacar como conclusión que es considerable usar *Deep Learning* cuando la precisión de los clasificadores clásicos no es suficiente (es el caso del *ImageNet*), cuando se disponga de una red entrenada que se pueda aprovechar o cuando haya una gran cantidad de categorías de imágenes (cientos, miles o más como puede ser en la tarea del reconocimiento de caras).

4. IMPLEMENTACIÓN SOFTWARE Y REALIZACIÓN DE PRUEBAS

En este apartado se va a comentar cómo ha sido todo el proceso de manejo de la máquina de soldadura (la cual se puede configurar automáticamente), gestión de los gases (tanto argón como aire) ... Posteriormente se recoge el espectro de dicha soldadura con una fibra óptica, y gracias al espectrómetro podremos visualizar como es el espectro con el *Software* que se maneja en el laboratorio. Tras esto, se comenzará a cambiar parámetros para simular todos los defectos que se van a tratar durante este trabajo.

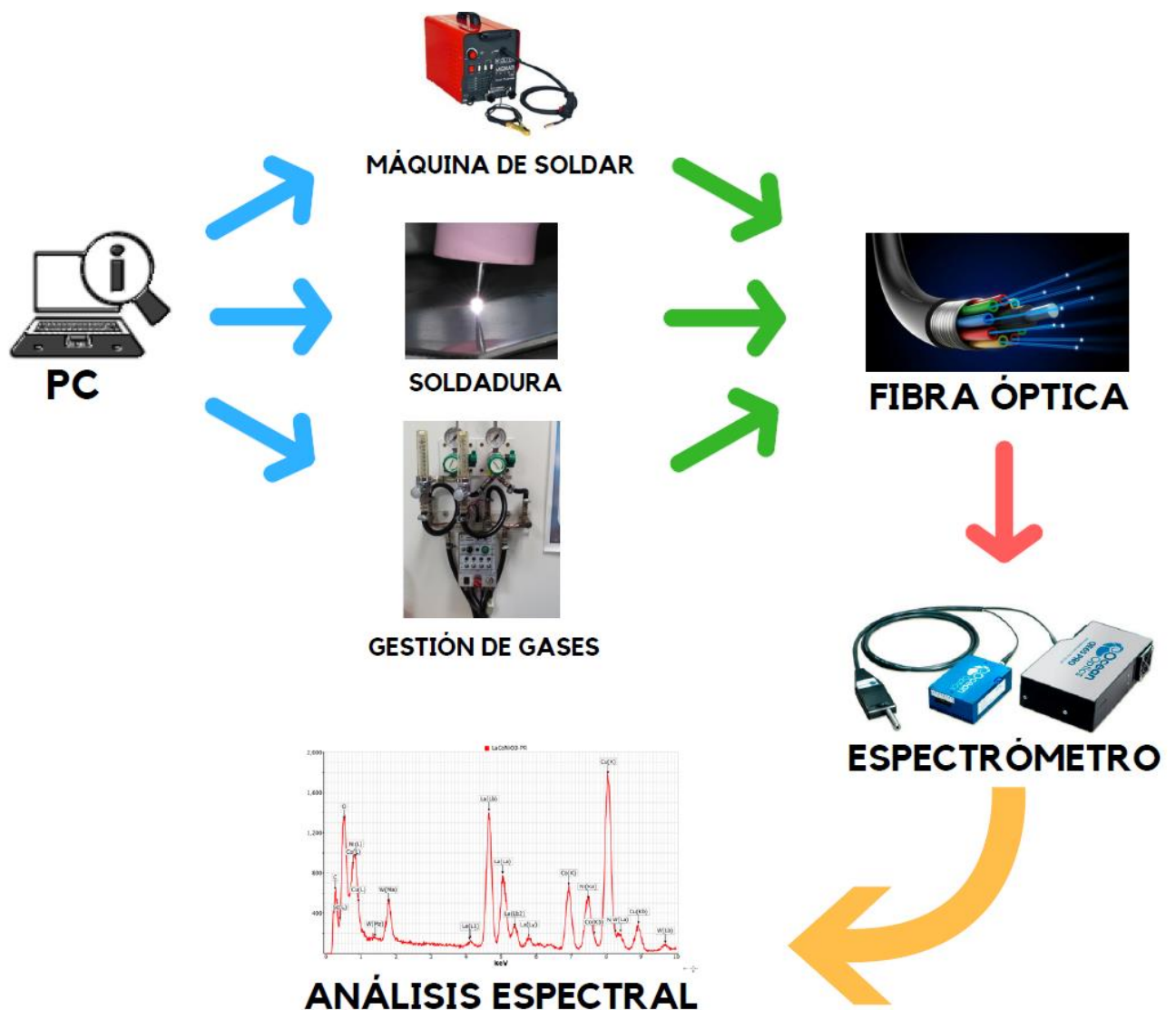


FIGURA 4.1 PROCESO A LLEVAR A CABO

Para realizar las distintas pruebas, se ha conectado a la antorcha una fibra óptica en forma de Y, para poder realizar las medidas con dos espectrómetros distintos (BRC112E-V y HR 4000) previamente calibrados. Esto se puede comprobar en la imagen 4.2.

La antorcha siempre debe estar en perfectas condiciones, es decir, tras varias soldaduras el electrodo se puede consumir y entonces hay que afilarlo.

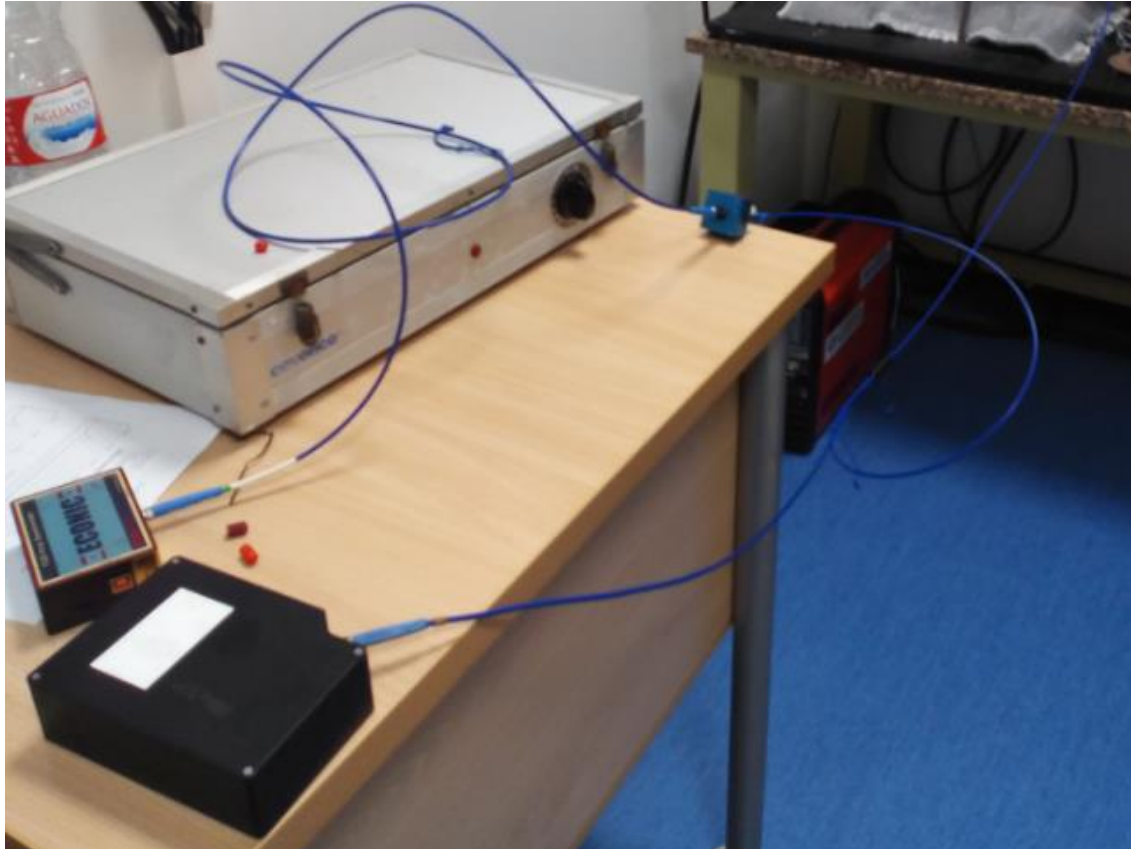


FIGURA 4.2 ESPECTRÓMETROS CONECTADOS EN FORMA DE Y

El procedimiento de calibrado se lleva a cabo manualmente, y aunque es un proceso sencillo, también es lento. Para ello se una conecta una fuente calibrada en longitud de onda, teniendo en cuenta que las líneas de emisión de dicha fuente son conocidas.

El proceso consiste en anotar dónde se encuentran los máximos de intensidad de los picos. Tras esto se relacionan los píxeles obtenidos con el *Software*, con las líneas de emisión de la fuente. Después se hace una regresión y se calcula el polinomio por el que hay que relacionar los píxeles, guardando los valores de los coeficientes en los valores del espectrómetro que salen en el software *MiSpectroMon*.



FIGURA 4.3 Lámpara de calibración

Cada espectrómetro será mejor dependiendo las características de cada soldadura, ya que el primero tiene un espectro de longitud de onda más amplio, por lo que se maneja más longitudes de onda, y el segundo, uno más corto, pero con más precisión en ese ancho espectral y muestra más líneas de emisión. El espectrómetro BRC112E-V es sensible en un rango de 300 a 850 nm aproximadamente, mientras que el HR 4000 de 200 a 1100 nm.

Una vez hecho esto, se usará el programa *WeldStorm 2.0* para realizar las pruebas de manera automática mediante el controlador. Con este *software* se puede realizar la soldadura de manera automática para facilitar el proceso, configurando los parámetros a usar en cada simulación. De este modo se manejan distintos valores de distancia para hacer el cordón de soldadura más o menos corto, la velocidad a la que la antorcha realizará la soldadura, intensidad de la misma, frecuencia, cortes de aire o argón...

Una vez fijados los anteriores parámetros, se puede añadir eventos para facilitar la soldadura. El evento *Change_Speed* permite en un intervalo previamente seleccionado, aumentar o disminuir la velocidad a la que la antorcha suelda. Otro parámetro importante en este trabajo es *Gas_Ar_Torch* y *Gas_Air_Torch*, ya que se podrá cortar el suministro de aire o de gas de protección (Argón). Los demás eventos no son usados, pero modifican la intensidad de soldadura, frecuencia, *Duty Cycle*...

Cuando se tenga conectado el controlador, se coloca la placa de acero inoxidable junto a la antorcha para iniciar las pruebas. Esta placa es puesta agarrada con tornillos para que al soldar no se mueva y produzca errores o se rompa el electrodo.

En la imagen 4.4 se muestra la antorcha para soldar, el electrodo, un tubo aspirador para recoger el aire y gas de la soldadura y la zona donde se colocará el material a soldar.



FIGURA 4.4 ELEMENTOS PARA REALIZAR LA SOLDADURA

A continuación, se van a explicar las distintas categorías de soldaduras que se han realizado para llevar a cabo el trabajo.

1. **Soldadura correcta:** condiciones habituales para realizar una soldadura entre dos chapas de acero.
2. **Soldadura con poca intensidad:** la corriente de la antorcha se disminuye un tanto por ciento de su valor habitual. También se puede producir aumentando la velocidad a la hora de soldar.
3. **Soldadura con mucha intensidad:** al contrario que la anterior, la corriente de la antorcha se ve aumentada. Del mismo modo, se puede bajar la velocidad a la que se mueve la antorcha para que incida más.
4. **Suciedad:** Se introducen materiales contaminantes en el proceso de soldadura. Esto se puede conseguir no limpiando la chapa de acero, poniendo celo, típex...
5. **Corte del gas de protección:** se corta el suministro de gas durante el proceso de soldadura. Esto se puede hacer durante un período concreto o si se quiere durante todo el proceso. En este caso, se hará en pequeños intervalos, para ver como disminuye la intensidad de las líneas de emisión del Argón.

Para comenzar, se hace una soldadura correcta, donde se ha usado una velocidad 9 mm/segundo e intensidad de arco de 85, 90 o 100 amperios, para tener un amplio rango de ejemplos de una buena soldadura.

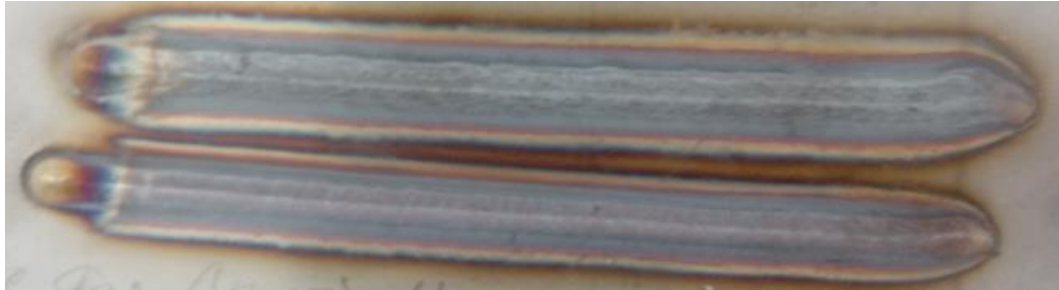


FIGURA 4.5 SOLDADURAS CORRECTAS

La siguiente tanda de soldaduras a realizar ha sido la de poca penetración, para la que se ha aumentado la velocidad a 15 mm/segundo en un primer ejemplo y después 18 mm/segundo o 20 mm/segundo, para que al soldar, se haga en menos tiempo la misma distancia y penetre menos el cordón de soldadura en la placa de acero inoxidable.



FIGURA 4.6 SOLDADURAS CON POCA INTENSIDAD DE PENETRACIÓN

De manera contraria, en la de excesiva penetración, la velocidad a usar ha sido de 6 mm/segundo en un principio, y posteriormente de 3 mm/segundo y 4 mm/segundo.

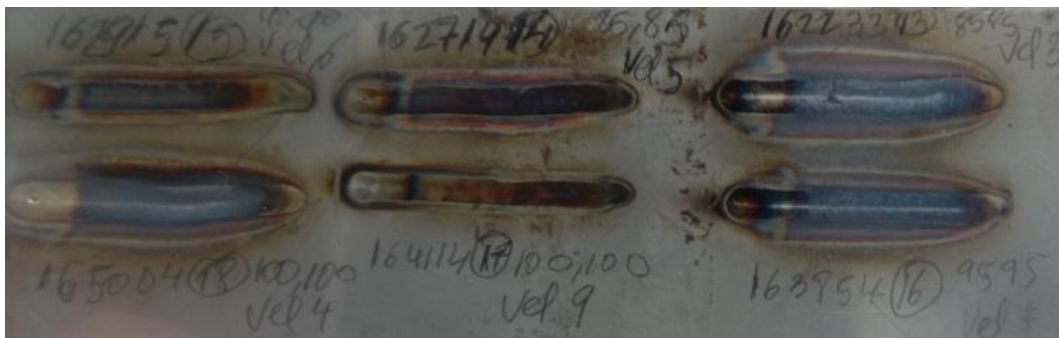


FIGURA 4.7 SOLDADURAS CON MUCHA INTENSIDAD DE PENETRACIÓN

El siguiente defecto que se ha querido simular ha sido el de suciedad en la soldadura, para lo que se han usado cinta adhesiva sobre la placa y después se ha despegado para que se quedara algún trozo de pegamento. Antes, se ha limpiado la placa con alcohol para eliminar otras posibles sustancias.



FIGURA 4.8 SOLDADURA CON SIMULACIÓN DE SUCIEDAD

La última simulación ha sido la de corte de gas, para la cual mediante el *WeldStorm 2.0*, se ha podido añadir un evento a la soldadura, para en un mismo proceso hacer tres cortes de gas.

En la siguiente imagen se muestra un cordón con un corte de gas, y en la parte inferior otros dos cortes de gas de argón en los intervalos (10,20) mm y (45,50) mm.



FIGURA 4.9 SOLDADURAS CON CORTES DE GAS

Los cortes de gas se han simulado mediante un evento, el cual manda una orden y corta la salida de aire/gas del elemento en cuestión. Dicho corte de gas no afecta al instante, ya que en el ambiente queda gas, el cual poco a poco va siendo aspirado, por eso en los espectros posteriores al corte de gas, siguen apareciendo líneas de emisión del argón, y poco a poco va disminuyendo la intensidad de las mismas.

En la siguiente figura se muestran las longitudes de onda (696nm, 706nm, 738nm, 750nm, 751nm, 763nm, 810nm) donde se encuentran las líneas de emisión más importantes del Argón, y en el espectro se ve que la intensidad en esas longitudes de onda está reducida.

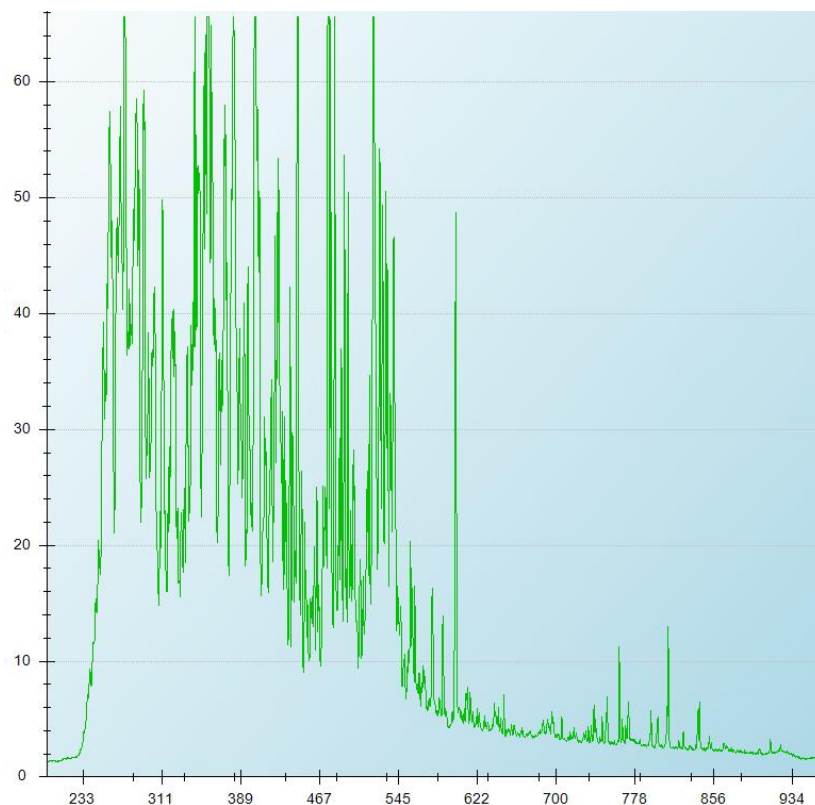


FIGURA 4.10 LAS LÍNEAS DE EMISIÓN DEL ARGÓN ESTÁN DISMINUIDAS

En la imagen 4.11 se muestra la máquina que da aire y de gas de protección (argón) a la soldadura, la cual se puede parar de manera automática manejando los controles.



FIGURA 4.11 MÁQUINA QUE DOTA CON AIRE Y ARGÓN

A continuación, se usa el programa *MiSpectroMon*, para ver los espectros de cada soldadura, donde ahí, se puede ver la composición de cada proceso. A lo largo del eje X de coordenadas se muestra el espectro que se recoge el espectrómetro gracias a la fibra óptica, mientras que en el eje Y, la intensidad de las líneas de emisión, indica los componentes químicos que aparecen en la soldadura.

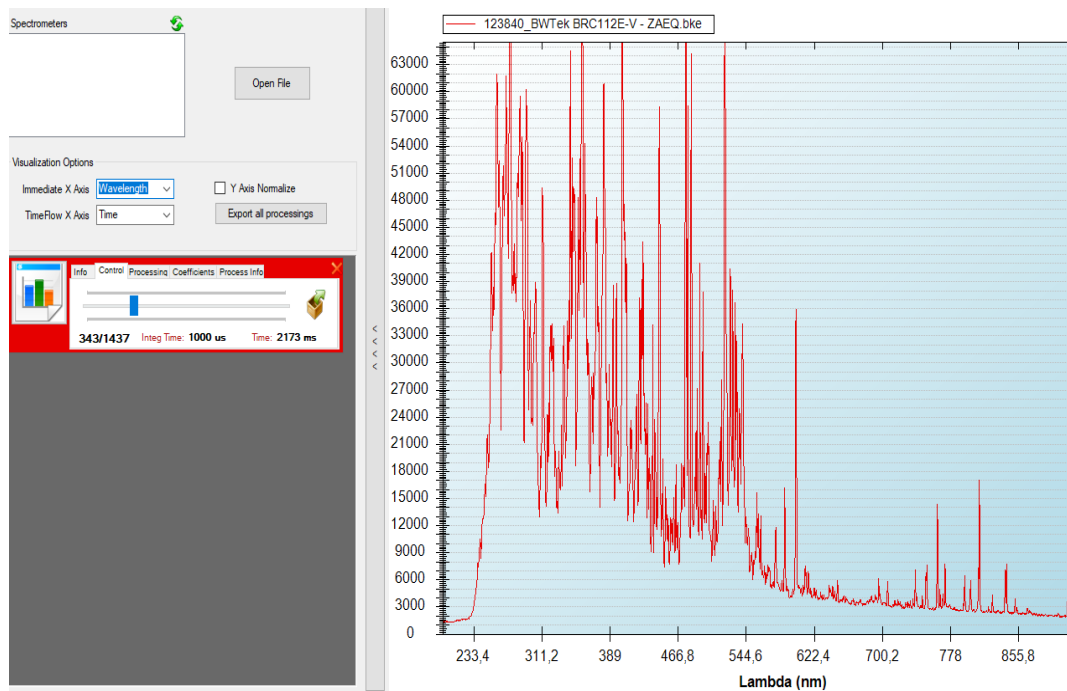


FIGURA 4.12 INTERFAZ DEL SOFTWARE *MiSpectroMon*

El extraer los distintos componentes químicos de la soldadura llevaría mucho tiempo, porque hay que acudir a la base de *Nist* para comparar los picos y sus relaciones de intensidad, para ver si son de ese compuesto químico en especial o son de otro.

Como este proyecto se ha enfocado al uso de una estrategia basada en *Deep Learning*, lo que se hace es pasar esos espectros .bke a imágenes .png mediante una aplicación creada con el programa *Visual Studio*.

Con un código más o menos corto, se crea una aplicación con tres botones distintos, como se muestra en la imagen 4.13.

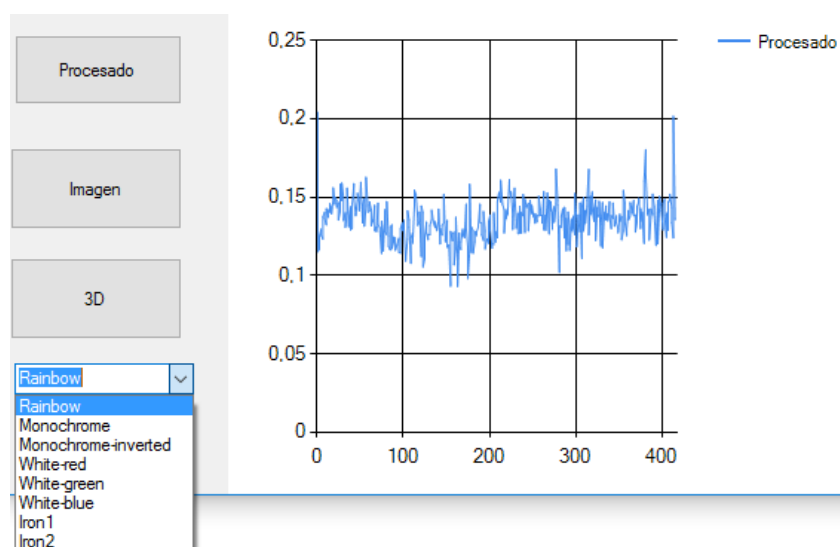


FIGURA 4.13 APLICACIÓN CREADA CON *VISUAL STUDIO*

El primer botón, es el que se nombra **procesado**. La función de este botón es coger los picos más cercanos y sacar una relación entre ellos. Después muestra el resultado por pantalla, tal y como se muestra a la derecha de la imagen 4.13.

El segundo botón **imagen**, guarda la gráfica del procesado en una carpeta definida en el código con formato .png.

El tercero es **3D**, el cual calcula el mapa de bits de la soldadura escogida. Con esto lo que se consigue es tener en una imagen la soldadura. Esta imagen sí podrá ser tratada mediante *Deep Learning* para entrenar la red.

La imagen se puede definir con distintos formatos de color, tal y como se puede ver en el desplegable de la imagen anterior. Esto se ha conseguido mediante un *Switch*, donde al elegir la paleta de color que se quiere para nuestra imagen, se accede a dicha función, y se ejecutará el *Case* que corresponda a ese valor. El *código C* de la paleta de colores (RAINBOW) se encuentra en el **ANEXO 2**.

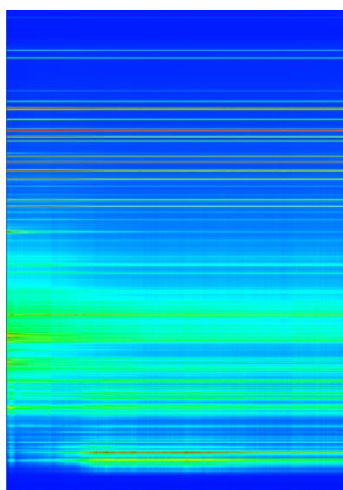


FIGURA 4.14 IMAGEN 3D HECHA CON LA PALETA RAINBOW

Para entrenar una red *Deep Learning* hacen falta miles de elementos de cada grupo a diferenciar, por eso en este momento, se llegó a la conclusión de que esta no era una buena manera de entrenar la red, ya que se dispone de 5 o 6 muestras para cada caso. Por eso se decide obtener todos los espectros de cada soldadura.

Primeramente, se cambia el código en el programa *Visual Studio* (figura 4.15) para conseguir que se recorra todo el cordón de soldadura (figura 4.16) y se vayan guardando automáticamente todos los espectros a lo largo de todo el proceso.

```

private void button2_Click(object sender, EventArgs e) // creo el botón para las imágenes y el chart
{
    if (openFileDialog1.ShowDialog() == System.Windows.Forms.DialogResult.OK)
    {
        string path = openFileDialog1.FileName;
        SpectrumUtils.FileStorage.BKEReader bke = new SpectrumUtils.FileStorage.BKEReader(path); //se lee el espectro

        for (int i = 0; i <= bke.PositionMax; ++i) //recorremos todos los espectros del corsón de soldadura
        {
            chart1.Series.Clear(); //limpiamos la la gráfica
            chart1.Series.Add("Procesado");
            chart1.Series[0].ChartType = System.Windows.Forms.DataVisualization.Charting.SeriesChartType.FastLine;
            chart1.Series[0].IsVisibleInLegend = false; //eliminamos la leyenda de la gráfica
            this.chart1.ChartAreas[0].AxisX.MajorGrid.LineDashStyle = 0; //eliminamos la subdivisión del eje X
            this.chart1.ChartAreas[0].AxisY.MajorGrid.LineDashStyle = 0; //eliminamos la subdivisión del eje Y
            this.chart1.ChartAreas[0].AxisX.LabelStyle.Enabled = false; //eliminamos los números del eje X
            this.chart1.ChartAreas[0].AxisY.LabelStyle.Enabled = false; //eliminamos los números del eje Y

            bke.Position = i;
            bke.Read();

            for (int j = 0; j < bke.SpectrometerInfo.NPixels; ++j)
            {
                chart1.Series[0].Points.Add(bke.CurrentSpectrum[j]); //pintamos sobre la gráfica el espectro i
            }
            chart1.Update();
            chart1.SaveImage(bke.File.FullName.Replace(".bke", "_" + i.ToString() + ".png"), ImageFormat.Png); //guarmamos la gráfica como imagen
        }
        bke.CloseFile(); //cerramos el espectro
    }
    return;
}

```

FIGURA 4.15 CÓDIGO C PARA CREAR LA APP CON VISUAL STUDIO

Así, de un solo cordón se pueden conseguir cientos de distintos espectros para cada prueba.



FIGURA 4.16 CORDÓN DE UNA SOLDADURA CORRECTA

De todos esos espectros hay que eliminar los primeros y los últimos. Para ello se abre el archivo .bke con *MiSpectroMon* para ver cómo van variando las líneas de emisión del espectro a lo largo del proceso. Al principio las líneas son débiles o incluso pueden saturar, lo mismo que ocurre al final, por lo que esas muestras son las que se decide eliminar. Además, en las pruebas con corte de gas, se eliminarán también todas las muestras fuera de los intervalos de ausencia del gas de protección. Esto se consigue con una sencilla regla de tres, sabiendo que, por ejemplo, el cordón de soldadura es de 8 cm, y se ha provocado un corte de gas en un intervalo [2.5, 5.0] centímetros.

En la imagen 4.17 se muestra la nueva solución de la aplicación creada, donde la imagen de cada espectro es más grande y sencilla, lo cual ayudará a la clasificación *Deep Learning*.

En las imágenes se muestra el espectro tal y como aparece en el programa *MiSpectroMon*. Además, se ha eliminado el *grid* de las imágenes y los valores de los ejes de coordenadas, para eliminar el “ruido” que añaden.

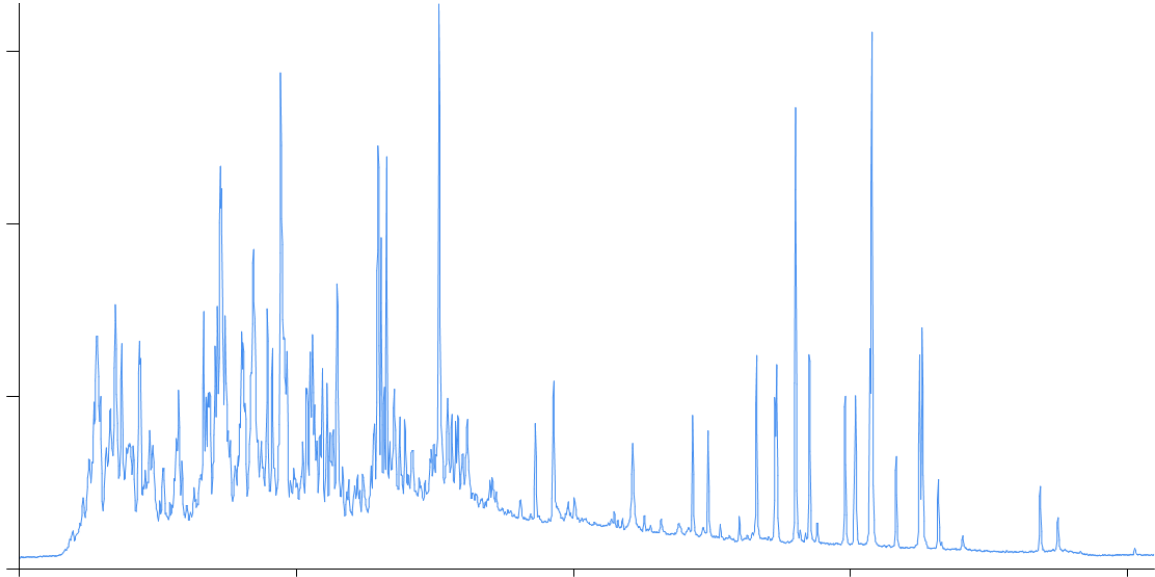


FIGURA 4.17 IMAGEN PARA ENTRENAR LA RED

Para continuar se ha instalado en el ordenador *Matlab* 2017, ya que en la nueva versión se incluyen características, funciones y *toolbox* específicas de *Deep Learning*.

5. RESULTADOS EXPERIMENTALES

En este apartado se va a comentar cómo se ha realizado la simulación en *Matlab* para las distintas categorías de imágenes, los parámetros para entrenar la red y los resultados que se han obtenido.

Indagando por Internet, se ha encontrado una *toolbox* de *Matlab* llamada *Alexnet*. Esta herramienta utiliza imágenes cotidianas (mesas, coches, vasos...) para entrenar la red y después dar la solución a los elementos que se ven por una cámara, etiquetándolos con el nombre que le corresponde. En el caso de este trabajo, esta *toolbox* no ha sido utilizada, porque se tienen imágenes ya clasificadas y además todas ellas semejantes, ya que son espectros.

Con el código Matlab que se puede ver en el **ANEXO 1**, se ha entrenado la red. El entrenamiento usa un algoritmo que es *descenso estocástico a la dirección del gradiente*. Esto se usa para definir las opciones de entrenamiento y tasas de aprendizaje para la red.

En el código utilizado se indica que las imágenes a tratar deben tener unos tamaños concretos, siendo de 80x60x3. El 80 y 60 indican los píxeles de la imagen y el 3, que la imagen es RGB, es decir, a color. En el caso de las imágenes extraídas con la aplicación *Visual Studio*, las imágenes tienen un tamaño de 1216x715x3, lo que indica que ha habido que redimensionar las imágenes para que puedan ser tratadas por la red neuronal. Con el siguiente código que se muestra en la imagen siguiente, se ha conseguido lo anterior.

```
a=dir('*.png');  
for k = 1:length(a)  
    c=imread(a(k).name);  
    cd=imresize(c,[60 80]);  
    imwrite(cd,sprintf('%i.png',k));  
end
```

FIGURA 5.1 CÓDIGO PARA REDIMENSIONAR LAS IMÁGENES

Lo que hace se consigue es coger todas las imágenes .png de un directorio y redimensionarlas al tamaño deseado, guardando los resultados con nombre desde 1.png hasta K.png, siendo *K* el número de imágenes que hay en dicho directorio. Una vez hecho esto ya se puede usar el código *Matlab* para proceder al entrenamiento.

El aprendizaje profundo requiere una gran cantidad de imágenes para alcanzar una alta precisión, reservándose un porcentaje de las mismas para posteriormente hacer el

proceso de testeo. En el caso con el que se trabaja, se reserva un 25 por ciento de imágenes para este cometido.

Primero se define la arquitectura de la CNN, donde se define la capa de entrada (*imageInputLayer*) con imágenes de 60x80 píxeles y 3 canales. A partir de aquí se combinan capas de tipo de convolución, *reluLayer* y *maxPooling2Layer*. A la capa de convolución se le pasa el tamaño del filtro, que es 15x15 o 15x16. Seguido están las capas *fullyConnectedLayer* (totalmente conectadas). Finalmente están las capas *softmaxLayer*, que traduce los valores de la última capa de neuronas y de probabilidades y después la capa de clasificación para etiquetar la imagen.

```
layers = [imageInputLayer([60 80 3]
convolution2dLayer(15 ,15);
reluLayer();
maxPooling2dLayer(3, 'Stride',3);
convolution2dLayer(5,16);
reluLayer();
maxPooling2dLayer(2, 'Stride',2);
fullyConnectedLayer(256);
reluLayer();
fullyConnectedLayer(5);
softmaxLayer();
classificationLayer()];
```

FIGURA 5.2 ARQUITECTURA DE LA CNN

En la siguiente imagen se adjunta una serie de parámetros que pueden ser cambiados para que el proceso de entrenamiento no sea demasiado largo, o simplemente que sea mejor.

```
%opts = trainingOptions('sgdm');
opts = trainingOptions('sgdm', ...
'InitialLearnRate', 0.001, ...
'LearnRateSchedule', 'piecewise', ...
'LearnRateDropFactor', 0.5, ...
'LearnRateDropPeriod', 8, ...
'L2Regularization', 0.004, ...
'MaxEpochs', 50, ...
'MiniBatchSize', 100, ...
'Verbose', true);
```

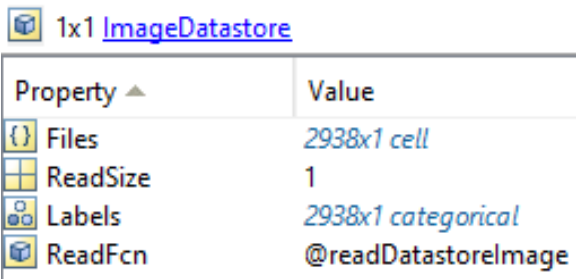
FIGURA 5.3 OPCIONES DE ENTRENAMIENTO

La variable *InitialLearnRate* indica la tasa con la que se comienza a aprender, la cual se irá reduciendo a la mitad, tal y como indica la variable *LearnRateDropFactor*. La variable *MinibatchSize* dice el número de imágenes que puedo pasar a la red de una vez durante el entrenamiento. Este número se puede adaptar en función del tamaño y

características de la GPU de la que se disponga. *MaxEpochs* indicará el número máximo de veces que vamos a pasar todo el conjunto de datos por la red , en este caso 50.

Una vez hecho esto se procede a llamar a la función de entrenamiento a la que le pasamos el conjunto de datos, la arquitectura (capas) y opciones definidas anteriormente. Como salida se obtiene la red neuronal y la información sobre el proceso de entrenamiento.

Para entrenar la red se clasifican las imágenes por categorías, teniendo en total el número de imágenes que se puede ver a continuación.



| Property ▲ | Value |
|------------|---------------------|
| Files | 2938x1 cell |
| ReadSize | 1 |
| Labels | 2938x1 categorical |
| ReadFcn | @readDatastoreImage |

FIGURA 5.4 IMÁGENES EN LA BASE DE DATOS

En la siguiente figura se ve el resultado del entrenamiento, donde se puede apreciar la gran cantidad de tiempo necesario para ello, algo más de 6000 segundos, ya que el tiempo de procesamiento depende de la potencia de cálculo y por tanto del número de GPUs que se tengan. El ordenador con el que se ha trabajado cuenta con 1 GPU.

| Epoch | Iteration | Time Elapsed (seconds) | Mini-batch Loss | Mini-batch Accuracy | Base Learning Rate |
|-------|-----------|---------------------------|--------------------|------------------------|-----------------------|
| 1 | 1 | 10.91 | 1.6595 | 8.00% | 0.0010 |
| 3 | 50 | 269.52 | 1.4589 | 44.00% | 0.0010 |
| 5 | 100 | 540.82 | 1.5996 | 32.00% | 0.0010 |
| 7 | 150 | 810.22 | 1.5793 | 39.00% | 0.0010 |
| 10 | 200 | 1108.65 | 1.5610 | 41.00% | 0.0005 |
| 12 | 250 | 1397.18 | 1.5630 | 31.00% | 0.0005 |
| 14 | 300 | 1676.01 | 1.5549 | 34.00% | 0.0005 |
| 16 | 350 | 1953.34 | 1.5418 | 34.00% | 0.0005 |
| 19 | 400 | 2231.76 | 1.5273 | 42.00% | 0.0003 |
| 21 | 450 | 2509.96 | 1.5499 | 31.00% | 0.0003 |
| 23 | 500 | 2787.84 | 1.5316 | 37.00% | 0.0003 |
| 25 | 550 | 3067.18 | 1.5657 | 31.00% | 0.0001 |
| 28 | 600 | 3341.81 | 1.4876 | 44.00% | 0.0001 |
| 30 | 650 | 3621.33 | 1.5400 | 32.00% | 0.0001 |
| 32 | 700 | 3891.82 | 1.5340 | 39.00% | 0.0001 |
| 35 | 750 | 4169.46 | 1.5247 | 41.00% | 6.25e-05 |
| 37 | 800 | 4442.36 | 1.5399 | 31.00% | 6.25e-05 |
| 39 | 850 | 4721.40 | 1.5359 | 34.00% | 6.25e-05 |
| 41 | 900 | 4995.55 | 1.5252 | 34.00% | 3.13e-05 |
| 44 | 950 | 5272.91 | 1.5116 | 42.00% | 3.13e-05 |
| 46 | 1000 | 5544.19 | 1.5418 | 31.00% | 3.13e-05 |
| 48 | 1050 | 5820.97 | 1.5229 | 37.00% | 3.13e-05 |
| 50 | 1100 | 6098.61 | 1.5631 | 31.00% | 1.56e-05 |

FIGURA 5.5 TABLA EN LA QUE SE MUESTRAN LOS RESULTADOS DEL ENTRENAMIENTO

Como se indicó en las opciones, el proceso acaba cuando se llega al número máximo de *Epochs*, para el cual se ha iterado unas 1100 veces. También se puede ver la precisión del proceso e entrenamiento, que se puede ver que es bajo.

El siguiente paso por seguir es llamar a la función *Classify*, a la que se le pasa la red ya entrenada y las imágenes que se habían guardado para el proceso de test. En este caso y como se indica en la siguiente imagen, el número de imágenes es 733.

```
>> testImages

testImages =

    ImageDatastore with properties:

        Files: {
            'C:\Users\vir\Desktop\Pruebas_final\BUENA\1.png';
            'C:\Users\vir\Desktop\Pruebas_final\BUENA\10.png';
            'C:\Users\vir\Desktop\Pruebas_final\BUENA\1000.png'
            ... and 730 more
        }
        Labels: [BUENA; BUENA; BUENA ... and 730 more categorical]
        ReadSize: 93
        ReadFcn: @readDatastoreImage

>> i

i =

    733
```

FIGURA 5.6 IMÁGENES GUARDADAS PARA EL PROCESO DE TEST

A continuación, se calcula la precisión total de la red como el número de imágenes que se etiquetan correctamente.

```
AciertoTotal =

    0.3438
```

FIGURA 5.7 PRECISIÓN TOTAL DE LA RED

En el siguiente apartado de este trabajo se pasará a comentar estos resultados y cómo se pueden mejorar.

6. CONCLUSIONES Y LÍNEAS FUTURAS

Como se ha comentado en anteriores apartados, gracias a *Deep Learning* se ha mejorado la precisión de acierto de Redes Neuronales Convolucionales en el etiquetado de imágenes o elementos que aparecen por pantalla. Para ello se necesita una gran cantidad de imágenes. En un principio, al tener 5 categorías distintas para entrenar la red, se pensó que era suficiente con 6 o 7 soldaduras de cada tipo, para posteriormente pasarlas a imagen 3D y entrenar la red. Después se pensó que no eran suficientes para el entrenamiento. Cada imagen recogía todos los espectros del cordón de soldadura, por lo que se intentó obtener todos esos espectros en distintas imágenes. Con esta solución se obtuvo unas 4000 imágenes distintas, de las cuales había que desechar algunas como se ha explicado anteriormente.

El acierto o no a la hora de etiquetar las imágenes depende de muchos factores, en este caso, el bajo acierto se puede deber al tamaño de las imágenes. Al principio estas imágenes eran grandes (1216x715 píxeles) y fácilmente diferenciables, pero al tener que redimensionarlas para poder alimentar la red, el tamaño de éstas se redujo bastante (80x60 píxeles), llegando casi a ser parecidas al abrirlas con el visualizador de fotos. Hay que tener en cuenta que con ese tamaño las imágenes sólo se diferencian por tener más o menos líneas de emisión con distinta intensidad en unas ciertas longitudes de onda.

Para intentar mejorar el 34.38 % de acierto, se han cambiado parámetros de las opciones de entrenamiento, poniendo por ejemplo el número de *Epochs* a 100, lo cual se tradujo en casi 4 horas de entrenamiento. El porcentaje de acierto sin embargo se mantuvo constante. EL cambio de parámetros es un proceso de prueba y error.

Como mejora en el futuro se puede tratar de incrementar el porcentaje de precisión de la red, tratando de buscar otro código/*toolbox* que permita tener un mayor porcentaje de acierto de la red, pudiendo aceptar imágenes de mayor tamaño para que la diferencia entre unos espectros y otros sea más apreciable. Por otro lado, se puede profundizar en el manejo de los parámetros, opciones, número de capas que se dan a la red, para ver si con esta serie de cambios se consigue una mejora.

Para ello sería bueno contar con un ordenador más potente que permita realizar las ejecuciones del código *Matlab* en menor tiempo, para poder realizar más pruebas.

En el ámbito de las soldaduras se podría realizar un mayor número de ellas para tener más información para dar a la red, simulando otro tipo de defectos como desalineación, otro tipo de suciedades, humedad... Para realizar las soldaduras hay que tener en cuenta que es bueno tener en todas ellas las mismas condiciones de trabajo, buen estado del electrodo, acero para la soldadura, gas de protección...

Otra mejora posible para realizar en el futuro podría ser que todo el proceso del trabajo se hiciera en un período de tiempo corto, desde el momento en que se produce la soldadura, la fibra óptica guía la luz al espectrómetro, para finalmente ver el espectro con el *Software*. Una vez se tengan los datos se entrenaría la red y se podría dar un resultado de la clasificación en un tiempo lo más rápido posible, todo esto sin necesidad de interacción de personas.

Como comentario final, lo que se ha intentado es reducir el tiempo y la dificultad de diferenciar las distintas categorías de soldadura, ya que en un principio hay que mirar las líneas de emisión del espectro y ver según las longitudes de onda e intensidad de las líneas de emisión qué componentes aparecen en la soldadura. Por eso, es interesante este trabajo, para que, en un futuro, no haya que realizar todo este proceso manualmente, si no que la red ya indique la categoría a la que corresponde la soldadura, facilitando y reduciendo el tiempo de clasificación.

7. BIBLIOGRAFÍA

7.1. REFERENCIAS DE TEXTO

- [1] <https://es.wikipedia.org/wiki/Soldadura>
- [2] <https://saludeestetica.org/la-importancia-de-la-soldadura-en-las-industrias/>
- [3] <http://construccionesmetalicascelr.blogspot.com.es/2011/11/control-de-calidad-en-soldaduras.html>
- [4] <https://es.slideshare.net/Orli10/ensayos-no-destructivos-aplicados-a-la-soldadura>
- [5] <http://www.interempresas.net/Medicion/Articulos/12133-Sistemas-de-control-y-monitorizacion-para-la-soldadura.html>
- [6] <http://www.twi-global.com/technical-knowledge/published-papers/laser-diode-based-vision-system-for-viewing-arc-welding-may-2009/>
- [7] http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1815-59442013000300009
- [8] <http://www.interempresas.net/MetalMecanica/Articulos/11149-Termografia-infrarroja-ensayo-no-destructivo-deteccion-defectos-componentes-aerospaciales.html>
- [9] Michael A. Graf, Víctor Benveniste, “*Real-Time Monitoring of Ion Source Plasma Using Optical Emission Spectroscopy*”
<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=812113>
- [10] https://physics.nist.gov/PhysRefData/ASD/lines_form.html
- [11] J. Mirapeix, A. Cobo, J.J. Valdiande, R. Ruiz-Lombera, P.B. García-Allende, O.M. Conde, L. Rodríguez-Cobo and J.M. López-Higuera, “*Spectroscopic Optical Sensors for Welding Diagnostics*”
<http://www.imeko.org/publications/tc4-2014/IMEKO-TC4-2014-492.pdf>
- [12] <http://revistasinvestigacion.unmsm.edu.pe/index.php/idata/article/view/12103/10829>
- [13] Li Dia, T. Srikanthanb, R.S. Chandelc, Inoue Katsunorid, “*Neural Network based self-organized Fuzzy Logic Control for Arc Welding*”
http://ac.els-cdn.com/S0952197600000579/1-s2.0-S0952197600000579-main.pdf?_tid=4ca4c66a-9c7b-11e7-90d7-00000aabb0f01&acdnat=1505744084_88683701cee5ff08777efac80e26760a
- [14] <http://www.tesisenred.net/bitstream/handle/10803/9441/tjimm1de1.pdf?sequence=1>
- [15] <http://www.it.uc3m.es/jvillena/irc/practicas/10-11/06mem.pdf>
- [16] <https://numenta.com/assets/pdf/whitepapers/hierarchical-temporal-memory-cortical-learning-algorithm-0.2.1-es.pdf>

[17] Luis Rodríguez-Cobo*, Ruben Ruiz-Lombera, Olga M. Conde, José-Miguel López-Higuera, Adolfo Cobo, Jesús Mirapeix, *"Feasibility study of Hierarchical Temporal Memories applied to Welding Diagnostics"*

[18] <http://cleverdata.io/machine-deep-learning/>

[19] <https://es.mathworks.com/videos/machine-learning-and-deep-learning-with-matlab-1499677542930.html>

7.2. REFERENCIAS DE IMÁGENES

FIGURA 1.2.1 PROCESO DE SOLDADURA

<http://www.interempresas.net/Medicion/Articulos/12133-Sistemas-de-control-y-monitorizacion-para-la-soldadura.html>

FIGURA 2.3.1 CONTROLADOR DEL LABORATORIO

Elaboración propia

FIGURA 2.3.2.1 ESPECTRO DE LA LUZ

<http://www.twi-global.com/technical-knowledge/published-papers/laser-diode-based-vision-system-for-viewing-arc-welding-may-2009/>

FIGURA 2.3.3.1 DISPOSICIÓN DE LA CÁMARA IR Y PIEZA DURANTE LA INSPECCIÓN POR TERMOGRAFÍA

http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1815-59442013000300009

FIGURA 2.3.3.2 EJEMPLO DE TERMOGRAFÍA PARA DETECCIÓN DE PATOLOGÍAS EN EDIFICACIÓN

<http://www.misecingenieria.cl/images/Camara1-2.jpg>

FIGURA 2.3.4.1 EJEMPLO DE UN ESPECTRO DE PLASMA EMITIDO DURANTE UNA SOLDADURA

Elaboración propia

FIGURA 2.3.4.2 PASOS A SEGUIR EN EL PROCESO DE USO DE UN ESPECTRO ÓPTICO DE PLASMA

<http://www.imeko.org/publications/tc4-2014/IMEKO-TC4-2014-492.pdf>

FIGURA 3.1 UNIONES ENTRE NEURONAS DENTRO DEL NOCÓRTEX HUMANO

<http://www.extremetech.com/wp-content/uploads/2015/07/neural-net-head.jpg>

FIGURA 3.1.1 ESTRUCTURA BÁSICA DE FUZZY LOGIC EN UN PROCESO DE SOLDADURA CON ARCO

http://ac.els-cdn.com/S0952197600000579/1-s2.0-S0952197600000579-main.pdf?_tid=337b55b6-9d41-11e7-9991-00000aacb360&acdnat=1505829082_c3401918eb89ea0984bbca049ebc4cec

FIGURA 3.1.2 ESTRUCTURA DE UNA RED NEURONAL BASADA EN EL CONTROL POR FUZZY LOGIC

http://ac.els-cdn.com/S0952197600000579/1-s2.0-S0952197600000579-main.pdf?_tid=337b55b6-9d41-11e7-9991-00000aacb360&acdnat=1505829082_c3401918eb89ea0984bbca049ebc4cec

FIGURA 3.2.1 ESTRUCTURA DE ACTUACIÓN DE LA NEURONA HUMANA, POR McCULLOCH Y PITTS

<http://www.tesisenred.net/bitstream/handle/10803/9441/tjimm1de1.pdf?sequence=1>

FIGURA 3.2.2 REPRESENTACIÓN DE UNA RED NEURONAL A DÍA DE HOY

<https://upload.wikimedia.org/wikipedia/commons/6/64/RedNeuronalArtificial.png>

FIGURA 3.3.1 DIAGRAMA DE 4 REGIONES ORGANIZADAS EN UNA JERARQUÍA DE 4 NIVELES

Diagrama simplificado de cuatro regiones HTM organizadas en una jerarquía de cuatro niveles, comunicando información dentro de cada nivel, entre niveles, y desde/hacia el exterior de la jerarquía

<https://numenta.com/assets/pdf/whitepapers/hierarchical-temporal-memory-cortical-learning-algorithm-0.2.1-es.pdf>

FIGURA 3.3.2 REDES DE DIFERENTES SENSORES CONVERGIENDO

<https://numenta.com/assets/pdf/whitepapers/hierarchical-temporal-memory-cortical-learning-algorithm-0.2.1-es.pdf>

FIGURA 3.3.3 ESPECTRO CAPTURADO DURANTE LA SOLDADURA, QUE ALIMENTA A LOS NODOS DE LA RED HTM

Luis Rodríguez-Cobo*, Rubén Ruiz-Lombera, Olga M. Conde, José-Miguel López-Higuera, Adolfo Cobo, Jesús Mirapeix, *“Feasibility study of Hierarchical Temporal Memories applied to Welding Diagnostics”*

FIGURA 3.4.1 GRÁFICA QUE MUESTRA EL INCREMENTO DE LAS TÉCNICAS MACHINE Y DEEP LEARNING

Interés a lo largo del tiempo. Búsqueda en la Web. Todo el mundo, 2004 – Hoy

<http://cleverdata.io/machine-deep-learning/>

FIGURA 3.4.2 RED NEURONAL ARTIFICIAL CON TRES CAPAS OCULTAS

<http://www.rsipvision.com/wp-content/uploads/2015/04/Slide5.png>

FIGURA 3.4.3 ESTRUCTURA BÁSICA DE UNA CNN

https://www.mathworks.com/content/mathworks/www/en/discovery/convolutional-neural-network/jcr:content/mainParsys/image_copy.adapt.full.high.jpg/1492406018870.jpg

FIGURA 3.4.4 CAPA DE CONVOLUCIÓN- ELECCION DE HIPERPARÁMETROS

<https://es.mathworks.com/videos/machine-learning-and-deep-learning-with-matlab-1499677542930.html>

FIGURA 4.1 PROCESO A LLEVAR A CABO

Elaboración propia

FIGURA 4.2 ESPECTRÓMETROS CONECTADOS EN FOMA DE Y

Elaboración propia

FIGURA 4.3 LÁMPARA DE CALIBRACIÓN

Elaboración propia

FIGURA 4.4 ELEMENTOS PARA REALIZAR LAS SOLDADURAS

Elaboración propia

FIGURA 4.5 SOLDADURA CORRECTA

Elaboración propia

FIGURA 4.6 SOLDADURA CON POCA INTENSIDAD DE PENETRACIÓN

Elaboración propia

FIGURA 4.7 SOLDADURA CON MUCHA INTENSIDAD DE PENETRACIÓN

Elaboración propia

FIGURA 4.8 SOLDADURA CON SIMULACIÓN DE SUDIEDAD

Elaboración propia

FIGURA 4.9 SOLDADURAS CON CORTES DE GAS

Elaboración propia

FIGURA 4.10 LAS LÍNEAS DE EMISIÓN DEL ARGÓN ESTÁN DISMINUÍDAS

Elaboración propia

FIGURA 4.11 MÁQUINA QUE DOTA DE AIRE Y ARGÓN

Elaboración propia

FIGURA 4.12 INTERFAZ DEL *SOFTWARE MiSpectroMon*

Elaboración propia

FIGURA 4.13 APLICACIÓN CREADA CON *VISUAL STUDIO*

Elaboración propia

FIGURA 4.14 IMAGEN 3D HECHA CON LA PALETA RAINBOW

Elaboración propia

FIGURA 4.15 CÓDIGO C PARA CREAR LA APLICACIÓN CON VISUAL STUDIO

Elaboración propia

FIGURA 4.16 CORDÓN DE UNA SOLDADURA

Elaboración propia

FIGURA 4.17 IMAGEN PARA ENTRENAR LA RED

Elaboración propia

FIGURA 5.1 CÓDIGO PARA REDIMENSIONAR LAS IMÁGENES

Elaboración propia

FIGURA 5.2 ARQUITECTURA DE LA CNN

Elaboración propia

FIGURA 5.3 OPCIONES DE ENTRENAMIENTO

Elaboración propia

FIGURA 5.4 IMÁGENES EN LA BASE DE DATOS

Elaboración propia

FIGURA 5.5 TABLA EN LA QUE SE MUESTRAN LOS RESULTADOS DEL ENTRENAMIENTO

Elaboración propia

FIGURA 5.6 IMÁGENES GUARDADAS PARA EL PROCESO DE TEST

Elaboración propia

FIGURA 5.7 PRECISIÓN TOTAL DE LA RED

Elaboración propia

ANEXO 1:

```
clear all
clc
```

```
%% Cargar los datos:
dataFolder = 'C:\Users\vir\Desktop\Pruebas_final';
% categories = {'Categorial', 'Categoria2', 'Categoria3'};
imds2 = imageDatastore(fullfile(dataFolder), 'IncludeSubfolders',
true, 'LabelSource', 'foldernames');
[trainingImages, testImages] = splitEachLabel(imds2, 0.75,
'randomize');
% To avoid weights randomly initialized, explicitly set the random
number generator seed.
rng('default')
% Los agrupamos en un único set de imágenes con la imágenes a tamaño
normal
```

```
%% Construimos la red.
layers = [imageInputLayer([60 80 3], 'Normalization', 'none');
convolution2dLayer(15, 15);
reluLayer();
maxPooling2dLayer(3, 'Stride', 3);
convolution2dLayer(5, 16);
reluLayer();
maxPooling2dLayer(2, 'Stride', 2);
fullyConnectedLayer(256);
reluLayer();
fullyConnectedLayer(5);
softmaxLayer();
classificationLayer()];
```

```
%opts = trainingOptions('sgdm');
opts = trainingOptions('sgdm', ...
'InitialLearnRate', 0.001, ...
'LearnRateSchedule', 'piecewise', ...
'LearnRateDropFactor', 0.5, ...
'LearnRateDropPeriod', 8, ...
'L2Regularization', 0.004, ...
'MaxEpochs', 50, ...
'MiniBatchSize', 100, ...
'Verbose', true);
```

```
net = trainNetwork(trainingImages, layers, opts);
```

```
%% Representacion de las imagenes.
figure;
montage((gather(net.Layers(2).Weights))); % Pesos de la red
title('Second Layer Weights');
% % Numero de filtros primera capa de convolucion = 15
% % Numero de filtros de la segunda capa de convolucion = 16
```

```

%% Testing the network
YTest = classify(net, testImages);



---



%% Convertimos las clases Pronosticadas de texto a numeros.
Etiquetas=zeros(length(testImages.Labels), 1);

for i=1:length(testImages.Labels);
    if testImages.Labels(i)=='SUCIEDAD';
        Etiquetas(i)=1;
    else if testImages.Labels(i)=='POCO';
        Etiquetas(i)=2;
    else if testImages.Labels(i)=='MUCHO';
        Etiquetas(i)=3;
    else if testImages.Labels(i)=='CORTEGAS';
        Etiquetas(i)=4;
    else if testImages.Labels(i)=='BUENA';
        Etiquetas(i)=5;
    end
end
end
end
end

Resultados=zeros(1, length(YTest));

Total1=0; Correctas1=0; Total2=0; Correctas2=0; Total3=0;
Correctas3=0; Total4=0; Correctas4=0; Total5=0; Correctas5=0;

for i=1:length(YTest);
    if YTest(i)=='SUCIEDAD';
        Resultados(i)=1;
        Total1=Total1+1;
        if Resultados(i)== Etiquetas(i)
            Correctas1=Correctas1+1;
        end

    else if YTest(i)=='POCO';
        Resultados(i)=2;
        Total2=Total2+1;
        if Resultados(i)==Etiquetas(i)
            Correctas2=Correctas2+1;
        end

    else if YTest(i)=='MUCHO';
        Resultados(i)=3;
        Total3=Total3+1;
        if Resultados(i)==Etiquetas(i)
            Correctas3=Correctas3+1;
        end

    else if YTest(i)=='CORTEGAS';
        Resultados(i)=4;
        Total4=Total4+1;
        if Resultados(i)==Etiquetas(i)
            Correctas4=Correctas4+1;
        end
    end
end

```

```

else if YTest(i)=='BUENA';
    Resultados(i)=5;
    Total5=Total5+1;
    if Resultados(i)==Etiquetas(i)
        Correctas5=Correctas5+1;
    end
end
end
end
end
end

Acierto1=Correctas1/Total1
Acierto2=Correctas2/Total2
Acierto3=Correctas3/Total3
Acierto4=Correctas4/Total4
Acierto5=Correctas5/Total5

AciertoTotal=(Correctas1+Correctas2+Correctas3+Correctas4+Correctas5)/
(Total1+Total2+Total3+Total4+Total5)

```

ANEXO 2:

```
public long EscalaColor(float Valor, int Tipo, long ValorMax, bool NoLineal) {  
    if( NoLineal ){  
        Valor = Convert.ToInt32((Valor * Valor * Valor) / (ValorMax * ValorMax));  
    }  
    long ValTemp;  
    switch(Tipo){  
        case 0: //paleta rainbow  
            ValTemp = Convert.ToInt32((1024 * Valor) / ValorMax);  
            if( ValTemp < 256 ){  
                EscalaColor = RGB(0, ValTemp, 255);  
                return;  
            }  
            if( ValTemp > 255 & ValTemp < 512 ){  
                EscalaColor = RGB(0, 255, 511 - ValTemp);  
                return;  
            }  
            if( ValTemp > 511 & ValTemp < 768 ){  
                EscalaColor = RGB(ValTemp - 512, 255, 0);  
                return;  
            }  
            if( ValTemp > 767 & ValTemp < 1024 ){  
                EscalaColor = RGB(255, 1023 - ValTemp, 0);  
                return;  
            }  
            EscalaColor = RGB(255, 0, 0);  
            return;  
        break;  
    }  
}
```
