

END-TO-END DEEP LEARNING FOR MALARIA DETECTION

**Tesis de Fin de Grado Entregada a la Facultad de
Escola Tècnica d'Enginyeria de Telecomunicació de
Barcelona**

Universitat Politècnica de Catalunya

por

Begoña Vizcaíno Gispert

**En parcial cumplimiento de los requisitos del grado de
Ciencias y Tecnologías de la Telecomunicación**

**Supervisoras: Elisa Sayrol Clois y Verónica Vilaplana
Besler**

Barcelona, Enero 2020

Abstract

Malaria is an infectious disease that affects between 300 and 500 million people a year in the whole world. Most cases appear in less developed countries, especially in Africa. In these places, the lack of resources delays diagnosis and consequently treatment. This project aims to develop a system of rapid detection of this disease using Deep Learning.

With this purpose, and using a database provided by Vall d'Hebron Hospital with images from the microscope of blood analysis from patients diagnosed with Malaria, a semi-automatic image labelling system has been developed and a neural network has been trained to detect Malaria infected cells and thus facilitate the diagnostic of the disease.

Resum

La Malària és una malaltia que afecta entre 300 i 500 milions de persones a l'any arreu del món, la majoria de casos es donen a països en vies de desenvolupament, sobretot a l'Àfrica i en aquests països la falta de recursos retarda el diagnòstic i per conseqüència el tractament. Aquest treball busca desenvolupar un sistema de detecció ràpida d'aquesta malaltia mitjançant Deep Learning.

Amb aquest propòsit i utilitzant una base de dades proporcionada per l'hospital Vall d'Hebron amb imatges d'un microscopi d'anàlisi de sang utilitzats per diagnosticar Malària s'ha desenvolupat un sistema d'etiquetatge d'imatges semiautomàtic i s'ha entrenat una xarxa perquè pugui detectar cèl·lules infectades de Malària i facilitar així el diagnòstic d'aquesta malaltia.

Resumen

La Malaria es una enfermedad infecciosa que afecta entre 300 y 500 personas en todo el mundo cada año. La mayoría de casos se suelen dar en países que están en vías de desarrollo, sobretodo en África y en estos países la falta de recursos retarda el diagnóstico y como consecuencia el tratamiento. Este trabajo busca desarrollar un sistema de detección rápida de esta enfermedad utilizando Deep Learning.

Con este propósito y usando una base de datos proporcionada por la Vall d'Hebron con imágenes de un microscopio en las que se analiza una muestra de sangre de un paciente al que se ha diagnosticado Malaria, se ha desarrollado un sistema de etiquetaje de imágenes semiautomático y se ha entrenado una red para que detecte células infectadas de Malaria y agilizar así el diagnóstico de esta enfermedad.

Agradecimientos

En primer lugar, me gustaría agradecer el soporte que me han dado mis tutoras Elisa y Verónica durante este trabajo, agradecerle también a Francesc Zarzuela, médico investigador en el hospital Vall d'Hebron por facilitarnos la base de datos de imágenes de frotis de gota fina de pacientes diagnosticados de Malaria y por compartir conmigo todo el conocimiento adquirido gracias a muchos años de investigación y de convivir de cerca con la Malaria.

Por último, agradecer todo el soporte sobre todo a nivel emocional de mis amigos, mi pareja y mis padres y hermanos a lo largo de este camino, durante la carrera y más concretamente durante la duración de este trabajo, sin ellos no habría sido posible.

Revision history and approval record

Revision	Date	Purpose
0	10/01/2020	Document creation
1	26/01/2020	Document revision
2	27/01/2020	Final version

DOCUMENT DISTRIBUTION LIST

Name	e-mail
Begoña Vizcaíno Gispert	begona.vizcaino@alu-etsetb.upc.edu
Verónica Vilaplana Besler	veronica.vilaplana@upc.edu
Elisa Sayrol Clois	elisa.sayrol@upc.edu

Written by:		Reviewed and approved by:	
Date	26/01/2020	Date	26/01/2020
Name	Begoña Vizcaíno Gispert	Name	Verónica Vilaplana Besler Elisa Sayrol Clois
Position	Project Author	Position	Project Supervisor

Índice

Figuras	8
Tablas	9
Introducción	
Malaria	10
Objetivos	11
Requerimientos y especificaciones	11
Métodos y procedimientos	12
Plan de trabajo y desviaciones	12
Estado del arte	13
Marco teórico	14
Malaria	14
Transmisión	
Diagnóstico	
Procesado de imagen	18
Watershed	
Operadores y filtros morfológicos	
Deep Learning	19
CNN	
RNN	
Fast R-CNN	
Faster R-CNN	
Metodología	28
Resultados	32
Presupuesto	35
Conclusiones y futuros desarrollos	36
Bibliografía	37
Anexos	38

Figuras

Figura 1: Países endémicos de Malaria en el 2000 y su situación en 2017. [WHO]	10
Figura 2: Extraída del Manual: Medios auxiliares para el diagnóstico de las infecciones palúdicas de la OMS	14
Figura 3: Ciclo Malaria CDC (https://www.cdc.gov/malaria/about/biology/index.html)	15
Figura 4 : Imagen extraída de: Manual of Basic Techniques for a Health Laboratory (2nd edition) de la OMS	16
Figura 5: Imagen extraída de: Laboratory Manual for Laboratory Technician Training	17
Figura 6: Imágenes de frotis de gota fina (izquierda) y gruesa (derecha).....	18
Figura 7: Watershed.....	19
Figura 8: Modelo simplificado de una neurona biológica	20
Figura 9: Arquitectura CNN	20
Figura 10: Funciones de activación	22
Figura 11: Precision y recall.....	24
Figura 12: Arquitectura R-CNN.....	24
Figura 13: Arquitectura Fast R-CNN	25
Figura 14: Arquitectura Faster R-CNN	26
Figura 15: Salida y funcionamiento RPN	26
Figura 16: Image Labeler Matlab.....	30

Tablas:

Tabla 1: Detección diferencial de la especie de parásito Plasmodium	16
Tabla 2: Evaluación de una red neuronal	23
Tabla 3: Distribución base de datos	29
Tabla 4: Valor de los parámetros del entrenamiento.....	32
Tabla 5: Resultados experimento 1.....	33
Tabla 6: Resultados del experimento 2.....	33
Tabla 7: Tabla de costes.....	35

1. Introducción

1.1. Malaria

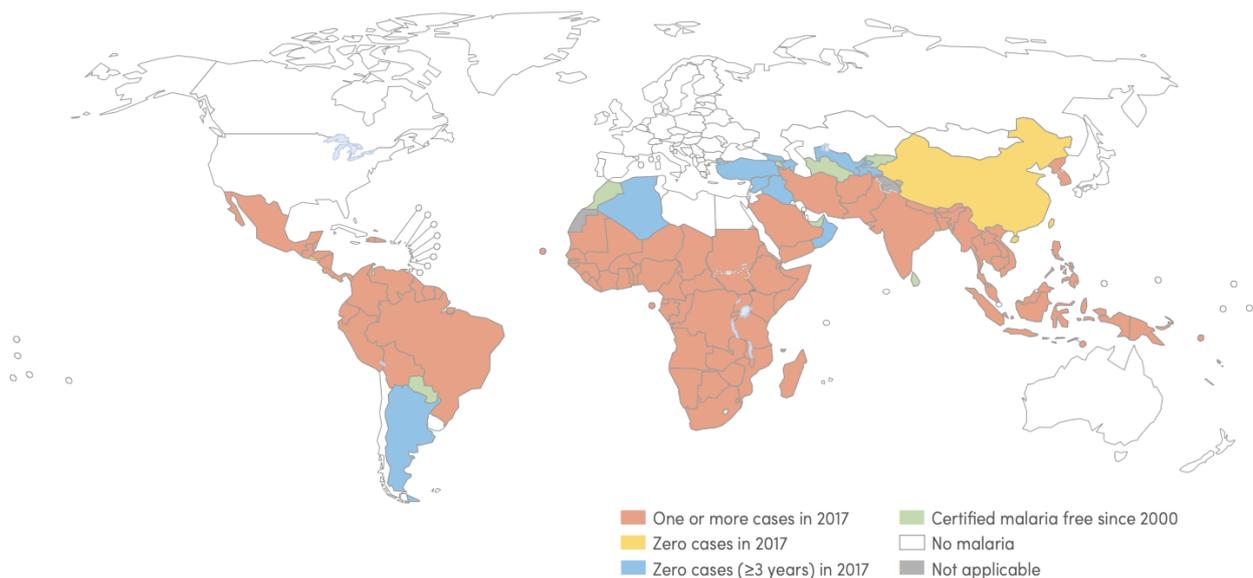
La **Malaria**, o **paludismo**, es una enfermedad infecciosa que afecta, cada año, entre 300 y 500 millones de personas, sobre todo en países en vías de desarrollo. Está causada por parásitos protozoarios del género *Plasmodium*, concretamente existen 4 especies de parásitos de esta familia importantes para el ser humano: *Falciparum* — la más común y la causante de la forma más grave de paludismo —, *Vivax*, *Ovale* y *Malariae*. El parásito, que llega a la sangre a través de la picada de un mosquito, ataca y destruye los glóbulos rojos y puede afectar a órganos vitales, llegando a causar problemas muy graves e incluso la muerte sobre todo cuando afecta al cerebro.

Este mapa, extraído del World Malaria Report 2018 realizado por la Organización Mundial de la Salud, muestra los casos de Malaria reportados en los distintos países en 2017. Los países con más afectación de Malaria, como decíamos, son países en áreas tropicales y subtropicales, sobretodo África y parte de Sudamérica.

África es la más afectada por una combinación de factores:

- La transmisión es alta a causa de la presencia del mosquito *Anopheles Gambiae complex*.
- La especie predominante es el *Plasmodium Falciparum*, que es la especie que tiende a provocar una malaria más dura y mortal.
- El clima tropical favorece la presencia de mosquitos durante todo el año.

La falta de recursos y la inestabilidad social no permiten hacer actividades de control y prevención del paludismo.



WHO: World Health Organization.

Figura 1: Países endémicos de Malaria en el 2000 y su situación en 2017. [WHO]

La voluntad de cambio del último de los factores mencionados en el párrafo anterior es el que origina este trabajo. Actualmente existen dos métodos para diagnosticar la Malaria, uno es muy rápido y sencillo, pero poco fiable, que es una prueba con **tira reactiva**; y el otro es muy preciso, pero requiere de más tiempo y técnicos expertos, ya que consiste en el **análisis microscópico** de una muestra de sangre del paciente para determinar si tiene o no parásitos de Malaria.

La posibilidad de obtener un diagnóstico rápido sin perder fiabilidad permitiría empezar antes los tratamientos y evitaría consecuencias peores, además de que ahorraría mucho tiempo a los expertos que, en estos países no suelen abundar.

Lo que nos planteamos inicialmente era desarrollar una solución que no solamente nos permitiera diagnosticar la Malaria sino que además determinara la especie concreta de parásito que estaba infectando al paciente, para poder darle el tratamiento más adecuado. En la actualidad, el diagnóstico diferencial se realiza mediante un frotis que llamamos de gota fina o extendida, mientras que la detección se realiza con el frotis de gota gruesa ya que este segundo tipo tiene más sangre y por tanto más concentración de parásitos por unidad de superficie y esto agiliza la detección.

Finalmente, no pudimos conseguir suficientes imágenes de gota gruesa por lo que decidimos desarrollar un método de diagnóstico de Malaria a partir de imágenes de frotis de gota fina, ya que como la detección es automática podemos analizar rápidamente varias imágenes de la misma extensión y determinar si el paciente tiene o no Malaria de la misma forma que lo haríamos manualmente con la gota gruesa.

De cara a futuros trabajos se podría extender el alcance este proyecto y detectar también especies.

1.2. Objetivos

Teniendo en cuenta este cambio los objetivos principales del proyecto son:

- a. Estudiar las redes neuronales convolucionales y sus usos en el ámbito médico.
- b. Aprender sobre el estado del arte en detección de Malaria u otras enfermedades infecciosas usando Deep Learning.
- c. Encontrar el mejor método para etiquetar las células para preparar las imágenes que se usarán para entrenar la red.
- d. Seleccionar una arquitectura de red neuronal con la que trabajar.
- e. Adaptar la red escogida correctamente para obtener la máxima exactitud.
- f. Probar el sistema diseñado y comparar los resultados con los obtenidos mediante otras técnicas.

1.3. Requerimientos y especificaciones

Para la realización de este proyecto ha sido necesario el etiquetaje de las imágenes proporcionadas por la unidad de medicina tropical del Hospital Vall d'Hebron y el uso de la base de datos Pascal VOC para pre entrenar la red y conseguir un buen resultado a pesar de no tener muchas imágenes.

Las imágenes se han etiquetado utilizando Matlab Image Labeler y para la detección posterior se ha realizado usando Keras sobre Tensorflow. Keras es una API de alto nivel de Tensorflow que permite construir y entrenar modelos de Deep Learning de forma sencilla y rápida.

Para aumentar la capacidad de procesado, la ejecución del entrenamiento de la red se ha realizado en el servidor Calcula, del grupo de Procesado de Imagen TSC.

1.4. Métodos y procedimientos

Para el etiquetaje se ha realizado una primera aproximación basada en métodos de procesado de imagen que se ha retocado usando la aplicación Image Labeler de Matlab.

La red utilizada para la detección en este caso de células infectadas de malaria es una Faster R-CNN desarrollada en Keras. La Faster R-CNN está formada por una RPN (Region Proposal Network) que propone las zonas con más probabilidad de que haya un objeto y una Fast R-CNN. La salida de la primera red se conecta a la entrada de la segunda para realizar la detección a partir de las regiones propuestas, en el apartado de marco teórico se explicarán estos conceptos con más detalle.

1.5. Plan de trabajo y desviaciones

En el apéndice se pueden encontrar las tareas, los hitos y el Gantt actualizado del proyecto. El plan de trabajo se ha modificado bastante respecto el que se había planeado originalmente debido a que se preveía entregar el trabajo en junio y finalmente se ha alargado hasta enero. Esto principalmente ha modificado las fechas de las últimas tareas e incluso se han añadido algunas tareas a medida que se ha ido desarrollando el trabajo.

2. Estado del arte

Las posibilidades que existen actualmente para diagnosticar la Malaria precisan de la presencia de expertos y requieren un tiempo que en algunos casos puede ser crucial para realizar un diagnóstico. La automatización de estos diagnósticos permitiría que los pocos médicos o especialistas que suele haber en los hospitales de los países más afectados por el paludismo pudieran determinar de forma rápida si un paciente tiene o no Malaria y darle así un tratamiento adecuado y que tendrá más efecto como antes se empiece a tomar.

Para ello, lo que se busca es automatizar el trabajo de estos especialistas, es decir, diseñar una “máquina” que sea capaz de analizar las muestras de sangre del paciente y determinar si este tiene o no Malaria. Para ello el Deep Learning es la solución perfecta, las redes convolucionales nos permiten extraer automáticamente características y aprender filtros para poder determinar si existe o no infección en nuestro caso. Para ello ni siquiera tenemos que programar las características diferenciales, solamente tendremos que darle a la red una base de datos con muchas imágenes y a partir de estas aprenderá. Además, si estas imágenes están etiquetadas por expertos y tenemos suficientes podremos llegar a igualar la precisión de estos expertos o incluso superarla en algunos casos.

Podemos plantear el problema de diagnóstico de formas distintas:

- Problema de **detección**: queremos detectar parásitos.
- Problema de **detección + clasificación**: queremos detectar las células y clasificarlas en infectadas o no infectadas.

Actualmente es posible encontrar artículos que enfocan el problema de formas distintas:

[1] Una red muy utilizada en problemas de detección es la **Retinanet**, esta red está formada por una pirámide de extracción de características sobre un modelo que suele ser ResNet50 y dos subredes, una de clasificación y otra de regresión.

[2] Otros ejemplos dividen el problema en dos, una segmentación y una clasificación. Usan redes de clasificación que han sido entrenadas con imágenes de células aisladas previamente y por consiguiente eliminan el problema de detección.

3. Marco teórico

3.1. Malaria

La **Malaria** es una enfermedad infecciosa causada por los parásitos de la familia Plasmodium. En los siguientes apartados se detalla su transmisión y diagnóstico.

3.1.1. Transmisión

La transmisión de esta enfermedad se produce a través de un mosquito, concretamente la hembra del mosquito **anopheles**. Este mosquito se infecta picando a una persona infectada y cuando pica a otra persona le infecta; el parásito pasa por diferentes estadios mientras está dentro del mosquito y después por distintos estadios también dentro del humano al que infecta, los primeros en el hígado humano y más adelante en el torrente sanguíneo. Empezaremos explicando los estadios del parásito dentro del mosquito:

1. El mosquito se infecta con parásitos del género *Plasmodium* cuando pica a una persona que tiene en la sangre la forma sexual del parásito (gametocitos).
2. Los **macro gametocitos femeninos** y **micro gametocitos masculinos** se transforman en **gametos maduros** en el intestino medio del mosquito.
3. La fecundación de un gameto femenino por un gameto masculino da lugar a un **cigoto** móvil (**ooquineto**) que emigra a la pared del estómago y ahí se convierte en **ooquiste**.
4. Este **ooquiste**, por división asexual genera en su interior hasta 10.000 **esporozoítos** alargados y fusiformes.
5. Cuando el **ooquiste** se rompe estos **esporozoítos** son liberados y se mueven por la cavidad corporal hasta llegar a las glándulas salivales.
6. Cuando el mosquito, ahora infectado, vuelve a picar para alimentarse con sangre, suelta los **esporozoítos** en la sangre de la persona a la que pica, por la sangre llegan al hígado, ahí invaden las células parenquimatosas, donde se transforman en **esquizontes exoeritrocíticos**.

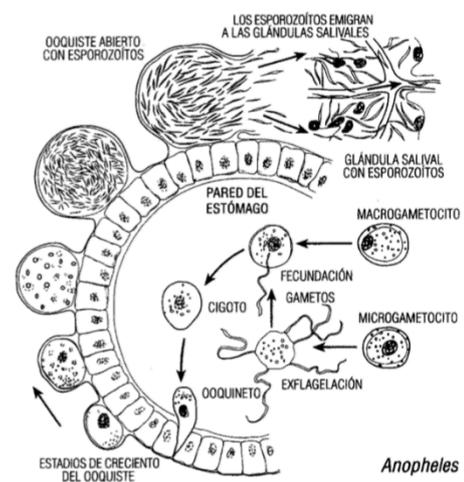


Figura 2: Extraída del Manual: Medios auxiliares para el diagnóstico de las infecciones palúdicas de la OMS

En el hígado es donde empieza a desarrollarse la enfermedad en los humanos:

1. Cuando la hembra del mosquito anopheles pica e inocula **esporozoítos** en el torrente sanguíneo del huésped humano, estos van por la sangre hasta el hígado, ahí se transforman en **esquizontes exoeritrocíticos** y siguen multiplicándose durante entre 5 y 15 días en función de la especie de *Plasmodium*.
2. Tras este período de tiempo, en la mayoría de casos, el **esquizonte** maduro explota, liberando miles de **merozoítos** en el torrente sanguíneo. Pero esto no es así en todos los casos, en el caso de *P. vivax* y *P. ovale*, algunos de los **esporozoítos** que habían llegado al hígado no se transforman inmediatamente en **esquizontes**, sino que producen **hipnozoítos**, que permanecen en estado latente durante meses en las células hepáticas. Si estos no son destruidos en el hígado mediante medicamentos antipalúdicos su posterior desarrollo dará lugar a recaídas.

3. En el torrente sanguíneo los **merozoítos** penetran en los glóbulos rojos y se nutren de la hemoglobina que estos les proporcionan, que les permite transformarse en **trofozoítos**.
4. Los **trofozoítos** se transforman en **esquizontes** que, de la misma forma que pasaba en el hígado, se van reproduciendo asexualmente hasta que, en los **esquizontes** maduros suele haber entre 6 y 24 **merozoítos** cuando explota.
5. Al romperse, los **merozoítos** son liberados al torrente sanguíneo e infectan otros glóbulos rojos dando lugar así al que llamamos **ciclo eritrocítico**, por el que se van infectando cada vez más glóbulos rojos y va aumentando la parasitemia.
6. Tras varias series de este ciclo algunos de los **trofozoítos** dan lugar a **microgametocitos** y **macrogametocitos** que, ingeridos por el mosquito hembra al picar al humano infectado, dan comienzo a otro ciclo de transmisión de la Malaria.

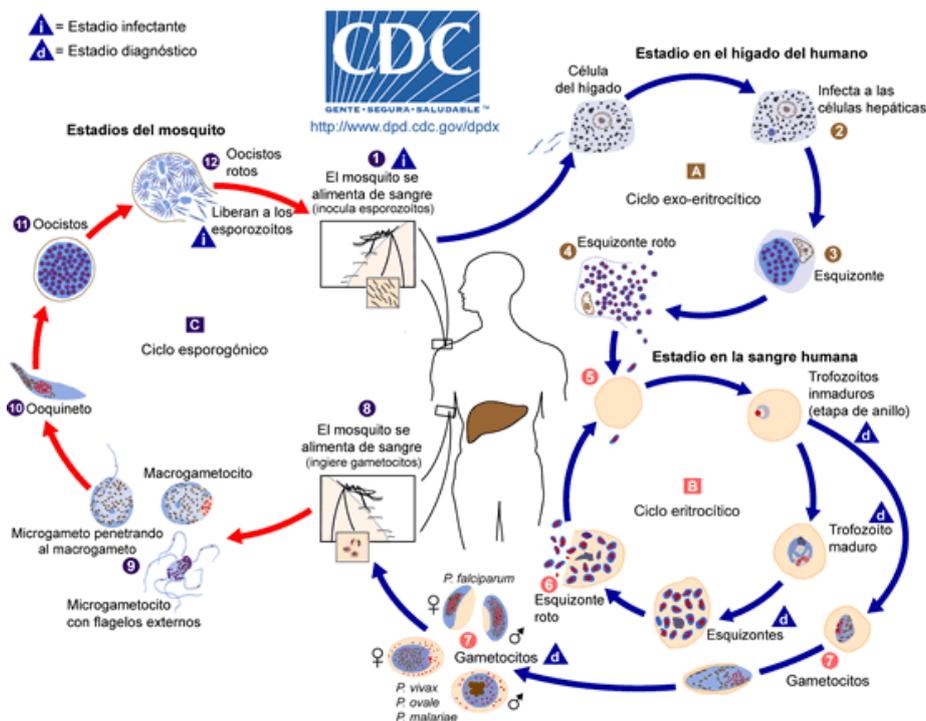


Figura 3: Ciclo Malaria CDC (<https://www.cdc.gov/malaria/about/biology/index.html>)

En función de la especie de parásitos estos estadios se manifiestan de formas ligeramente distintas, esto nos permite, en el diagnóstico, poder distinguir la especie de parásito con la que nos encontramos. Normalmente cuando se dan síntomas y por consiguiente cuando se suele realizar el diagnóstico es durante el **ciclo eritrocítico** y las formas que más solemos observar son los **trofozoítos**, los **esquizontes** y los **gametocitos**.

Las diferencias entre especies se encuentran en la forma del parásito, pero también en la forma del glóbulo rojo, ya que en algunos estadios el parásito modifica el tamaño y la forma de la célula e incluso genera distintos tipos de granulaciones en el citoplasma del hematocito parasitado.

	Plasmodium falciparum	Plasmodium vivax	Plasmodium malarie	Plasmodium ovale
Tamaño hematie parasitado	Normal 	Aumentado 	Normal 	Aumentado, tendencia oval y desflechado. 
Granulación citoplasma hematie parasitado	Puede presentar manchas de Maurer. 	Granulaciones azuladas de Schuffner 	No 	Granulaciones azuladas de Schuffner 
Nº plasmodios por hematie parasitado	1-4	1-2 (raramente)	1	1
Puntos de cromatina del trofozoíto	1-2	1-2 (raramente)	1	1
Forma trofozoítos jóvenes	Anillo delgado y pequeño 	Anillos deformes 	Anillos gruesos 	Anillos deformes 
Forma trofozoítos maduros	No frecuente en sangre periférica	Anillo grande irregular 	Anillo compacto redondo. Forma banda 	Anillo grande redondo compacto 
Esquizontes	No frecuente en sangre periférica	12-18 merozoitos	8-10 merozoitos	6-12 merozoitos
Forma gametocitos	Falciforme 	Redondeado 	Redondeado 	Redondeado 

Tabla 1: Detección diferencial de la especie de parásito Plasmodium

3.1.2. Diagnóstico

Actualmente existen dos formas de detectar la Malaria, una es más rápida y sencilla, pero menos precisa y la otra lleva más tiempo y requiere de personas expertas, pero es mucho más precisa.

- **Prueba con tira reactiva:** esta prueba es mucho más rápida y sencilla, conlleva en total unos 10 minutos y consiste en, mediante el uso de una tira reactiva, detectar anticuerpos monoclonales de la proteína (HRP-II) que suele estar presente durante la el estadio de reproducción asexual y los primeros estadios de gametocito del parásito.

La tira se prepara con un anticuerpo monoclonal de ratón, para simular el anticuerpo que generaría un ser humano, que se aplica en una línea transversal a aproximadamente 1 cm de su base. Una segunda línea punteada de antígeno HRP-II se incorpora en la varilla unos 2-3 mm por encima de la línea de anticuerpos monoclonales como control reactivo positivo.

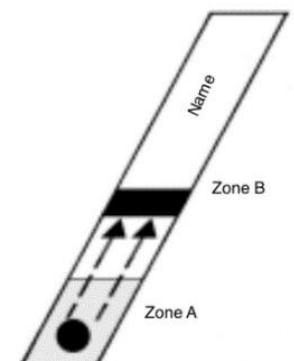


Figura 4 : Imagen extraída de: Manual of Basic Techniques for a Health Laboratory (2nd edition) de la OMS

Este método de diagnóstico consiste en:

1. Extraer unas gotas de sangre del dedo del paciente.
2. Introducir una gota en un tubo de ensayo que contenga reactivo de lisado.
3. A continuación, una gota de esta sangre lisada se coloca en uno de los agujeros del banco de reacción.
4. En ese mismo agujero se coloca la tira reactiva encima de la sangre lisada hasta que esta absorba toda la sangre.
5. Aplicamos una gota de reactivo de detección en la base de la tira, este reactivo actuará como marcador.
6. Cuando haya sido absorbido aplicamos dos gotas de reactivo de lavado para eliminar la sangre lisada.
7. Si el resultado es positivo una línea roja aparecerá en la tira reactiva (la línea que se había aplicado de anticuerpo), si es negativo solo se podrá ver la línea punteada de control.

Es una prueba muy rápida y sencilla, pero los estudios determinan que tiene una sensibilidad de entre el 86 y el 95% respecto al análisis microscópico llevado a cabo por técnicos expertos.

- **Análisis microscópico:** esta es la prueba realmente más fiable y más utilizada para determinar si un paciente padece o no Malaria. Consiste en analizar microscópicamente una muestra de sangre del paciente y ver si contiene o no parásitos *Plasmodium*. Esta prueba se puede realizar con frotis de dos tipos, de **gota fina** o de **gota gruesa**, se suelen realizar en el mismo porta objetos, pero es muy importante que no se toquen porque tienen fijaciones y extensiones distintas. El procedimiento para preparar las muestras para el análisis se puede dividir en 3 pasos, extracción, fijación y tinte:

EXTRACCIÓN: En el mismo porta objetos ponemos a un lado 3 gotas pequeñas de sangre (para la gota gruesa) y al otro una gota más grande, que sacaremos del dedo del paciente. El primer frotis a preparar es el de **gota fina** o extendida, su nombre se debe a que para prepararlo lo que debemos hacer es extender la gota más grande usando otro porta-objetos y deslizándolo por encima del primero. Una vez este está preparado y seco entonces preparamos el frotis de **gota gruesa**, para ello simplemente tenemos que mezclar bien las 3 gotas de sangre.

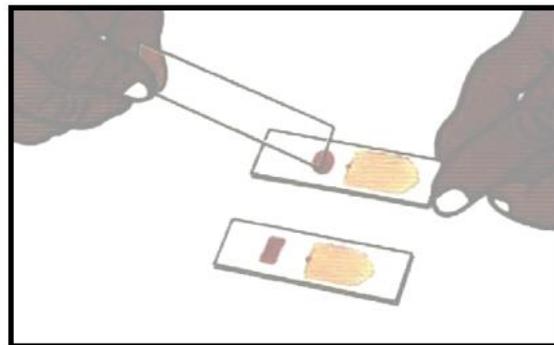


Figura 5: Imagen extraída de: Laboratory Manual for Laboratory Technician Training

FIJACIÓN: El tinte contiene **agua**, lo que, en contacto con los glóbulos rojos, los rompe, en el caso del frotis de gota gruesa queremos que se rompan las células y que los parásitos queden sueltos, pero en la **gota fina** queremos que los glóbulos rojos queden intactos lo que nos permitirá determinar la especie concreta de parásito *Plasmodium*. Para fijar lo único que debemos hacer es sumergir la parte del porta-objetos donde tenemos el frotis de **gota fina** en alcohol durante entre 3 y 5 segundos, después solo faltará dejarlo secar antes de teñir.

TINTE: El tinte que se suele utilizar es tinte de *Giemsa*, que contiene 3,5g de polvo de Giemsa, 250ml de metanol y 250ml de glicerol. Como tiene alcohol es importante diluirlo para evitar que fije los glóbulos rojos también en la **gota gruesa**.

El frotis de **gota gruesa** se utiliza para la detección porque contiene más sangre, lo que suele implicar más parásitos, y más facilidad para encontrarlos cuando hay pocos. El de **gota fina** en cambio tiene una sola capa de células, y los parásitos estarán dentro de los glóbulos rojos. Esto nos permitirá determinar la especie por dos motivos, el primero es que los parásitos estarán más ampliados gracias a la extensión, el segundo es que al estar dentro del hematíe podremos identificar también como le afecta y como ya hemos visto esto también es un factor clave para el diagnóstico diferencial.

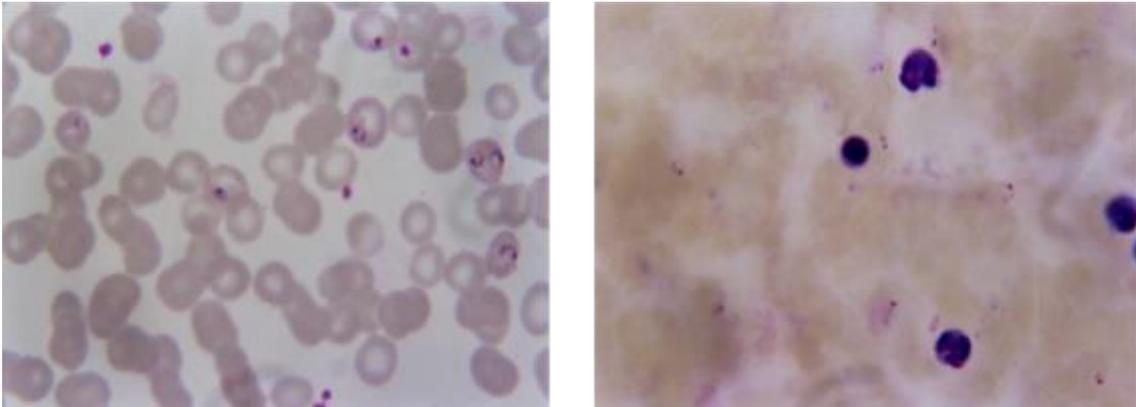


Figura 6: Imágenes de frotis de gota fina (izquierda) y gruesa (derecha)

3.2. Procesado de imagen

3.2.1. Watershed

La técnica de **Watershed** es una técnica de segmentación basada en marcadores. La filosofía de este método se basa en entender la imagen como una superficie topográfica en la que encontramos “cuencas” y “crestas” considerando el nivel de gris de un píxel como su altura en relieve. El algoritmo consiste en empezar a llenar de agua de distinto color (etiquetas) cada cuenca (mínimo), cuando se llega a algún pico, si se sigue llenando, el agua de distintas regiones se fusionará, por lo que se tienen que construir barreras que lo eviten, a medida que vaya subiendo más el agua se irán creando más barreras hasta que todos los picos queden bajo el agua. Estas barreras son la representación de la segmentación de la imagen. En la práctica, el método a seguir para realizar la segmentación consiste en marcar la región considerada objeto o primer plano para diferenciarlo del fondo y aplicar la función Watershed de Matlab que encuentra los límites de las regiones marcadas.

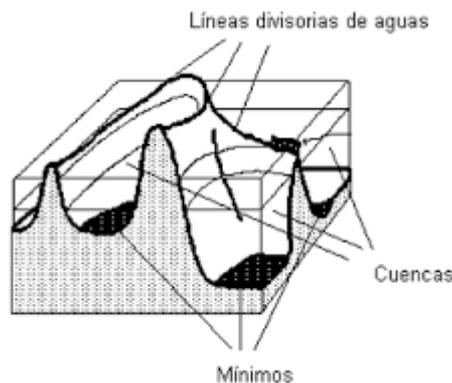


Figura 7: *Watershed*

3.2.2. Operadores y filtros morfológicos

En morfología matemática existen distintos operadores que cumplen el principio de superposición lineal y que permiten descomponer o crear señales. Pueden ser usados para modificar los elementos de una imagen. Los operadores básicos son el supremo y el ínfimo, y combinando estos operadores se crean otros operadores naturales que cumplen:

- Mantienen la estructura del retículo.
- Son compatibles con el principio de superposición.
- Son invariantes a la translación.

En este trabajo se han usado los operadores para realizar la marcación previa al watershed, concretamente:

- **Dilatación**: por cada punto, cálculo del mínimo nivel de gris debajo del elemento estructurante. En la práctica, permite eliminar componentes claros y ensanchar los elementos oscuros.
- **Erosión**: por cada punto, cálculo del máximo nivel de gris debajo del elemento estructurante traspuesto. En la práctica, permite eliminar componentes oscuros y como consecuencia separar objetos y evitar superposiciones.

Combinando dilataciones y erosiones entre ellas se obtienen los filtros morfológicos:

- **Opening** morfológico: Sean A y B dos imágenes, la **apertura** de A por B se obtiene de calcular la erosión de A por B y realizar la dilatación de la imagen resultante por B. En la práctica redondea los objetos y elimina bordes afilados.
- **Closing** morfológico: Sean A y B dos imágenes, el **cierre** de A por B consiste en la dilatación de A por B seguida por la erosión de la imagen obtenida por B.

3.3. Deep learning

El Deep Learning es un conjunto de algoritmos de aprendizaje automático que intenta modelar abstracciones de alto nivel en datos a partir de la creación de arquitecturas computacionales cuyo diseño está basado en las redes neuronales.

Una neurona, esta está formada por el soma o núcleo, que está rodeado de dendritas, el soma se conecta a las sinapsis a través del axón y la sinapsis a su vez se conecta a las dendritas de otras neuronas.

El funcionamiento de la red neuronal consiste en que a través de las dendritas llegan impulsos nerviosos provenientes de otras neuronas, cuando este impulso eléctrico supera un umbral, envía un voltaje a través del axón que llegará a las neuronas que estén conectadas a ella.

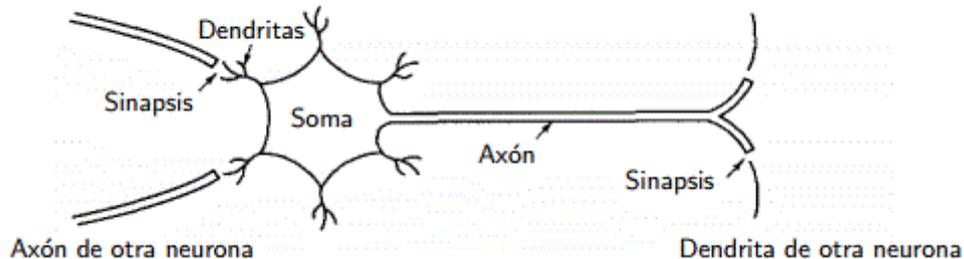


Figura 8: Modelo simplificado de una neurona biológica

En las redes neuronales artificiales se simula el funcionamiento de la neurona con la función de activación: de la misma forma que el soma manda un impulso cuando la combinación de los impulsos eléctricos que llegan a través de las dendritas supera un umbral, en las redes neuronales la función de activación determina el umbral que sirve para distinguir entre clases o entre la decisión de si es o no es un objeto.

3.3.1. CNN - Red Neuronal Convolutiva

Una **Red Neuronal Convolutiva** (CNN - Convolutional Neural Network) es un tipo de Red Neuronal en la que las neuronas están distribuidas en capas de filtros convolutivos de una o más dimensiones, lo que permite procesar datos multidimensionales como las imágenes. Después de cada capa se suele añadir una función para realizar una transformación no lineal. Las principales diferencias entre este tipo de redes neuronales y el resto de redes neuronales básicas son:

- Los pesos de cada capa son filtros que se aplican (convolución) directamente a la imagen.
- No cumple la propiedad de conectividad completa, es decir, cada neurona no está conectada a todas las neuronas de la capa anterior; esto permite que cada neurona se especifique en una región de la capa anterior y reduce el número de pesos y multiplicaciones necesarias.

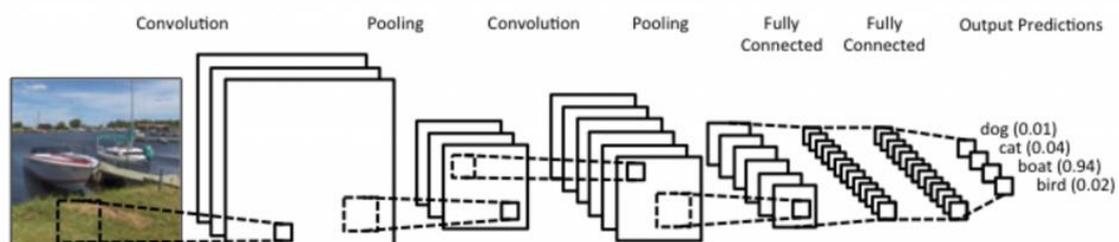


Figura 9: Arquitectura CNN

ESTRUCTURA Y FUNCIONAMIENTO

La estructura de una CNN consiste en múltiples capas de filtros convolutivos de una o más dimensiones, después de cada capa se añade una función para realizar una transformación no-lineal. Se podría considerar que hay dos fases, la de extracción de

características y la de clasificación, la primera está formada por capas convolucionales y de submuestreo intercaladas y la fase de clasificación está constituida por capas Fully Connected:

- Capas convolucionales: la **convolución** lo que permite es filtrar la imagen de entrada de manera que transforma los datos haciendo que ciertas características se vuelvan más dominantes en la imagen de salida, asignando a los píxeles que representan estas características unos valores más altos, dándoles más peso. A medida que los datos progresan por estas capas se disminuye su dimensionalidad de forma que las neuronas de las capas más lejanas son menos sensibles a perturbaciones de los datos de entrada, pero a la vez esto implica que se activan para características cada vez más específicas. La salida de cada capa se calcula mediante la siguiente fórmula.

$$Y_j = g \left(b_j + \sum_i K_{ij} \otimes Y_i \right)$$

- Capa de **submuestreo** o **pooling**: estas capas se intercalan entre las capas convolucionales para reducir la dimensionalidad de los mapas de activación resultantes de cada filtro. Originalmente esta operación se realizaba mediante un proceso de subsampling (sub-muestreo), pero actualmente se ha demostrado que operaciones como el Max-Pooling son mucho más eficaces. La operación de Max-Pooling consiste en encontrar el valor máximo dentro de una ventana de muestras y pasar ese valor como resumen de esa área, esto nos permite reducir el tamaño de los datos por un factor igual al tamaño de la ventana que habíamos definido.
- Capas **Fully-Connected** o de **clasificación**: después de una o más fases de extracción de características los datos llegan finalmente a la fase de clasificación. Hasta ese momento han sido depurados de una serie de características únicas para la imagen de entrada y en esta última capa, mediante unas funciones de activación, se clasifican las características extraídas hacia una etiqueta u otra, en función de los objetivos del entrenamiento. Se podría decir que las capas convolucionales y de reducción de muestreo extraen características y las capas fully-connected, que tienen todas las neuronas de la capa anterior conectadas, toman decisiones en base a estas características. Las funciones de activación más habituales son:
 - **Rectified Linear Unit (ReLU)**: anula los valores negativos del mapa de activación.
 - **Softmax** o **sigmoide**: permite crear una distribución de probabilidad sobre las posibles predicciones, aplasta la salida entre 0 y 1.

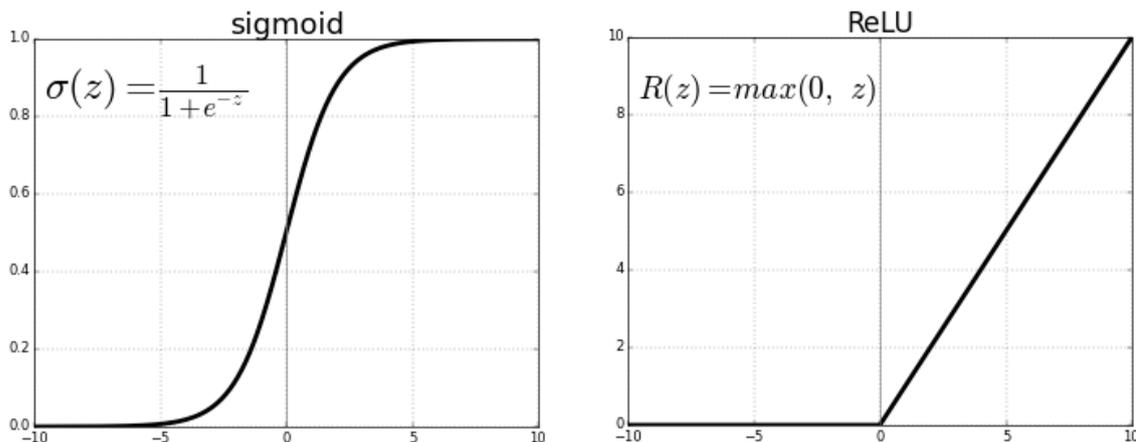
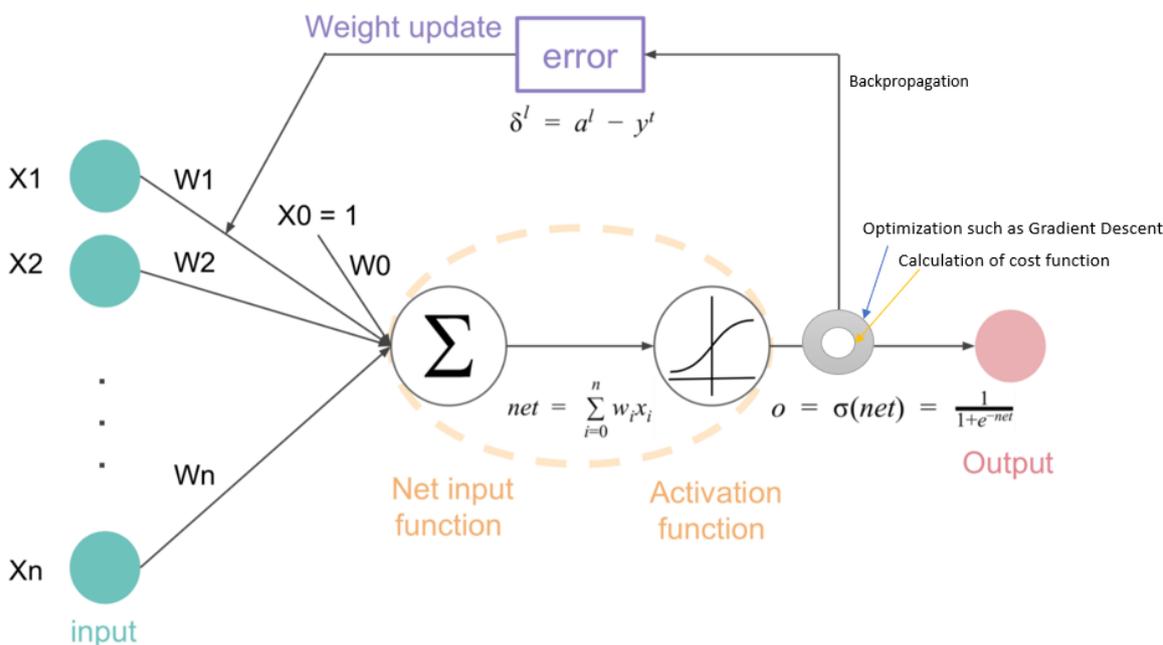


Figura 10: Funciones de activación

ENTRENAMIENTO DE LA RED

El entrenamiento es el proceso que consigue, dada una función de coste, encontrar la solución óptima (pesos y parámetros del modelo) que minimice el error de predicción sobre los datos de entrenamiento.



La mayoría de algoritmos se basan en el método del gradiente o “**gradient descent**”: en cada punto de decisión se calcula el gradiente y el error y se avanza en el sentido del gradiente negativo, para llegar al mínimo error. Para este cálculo sucesivo del gradiente se suele realizar “back-propagation” lo que significa que se empieza a calcular el gradiente por el final y en caso de detectar un error se vuelve atrás y se actualizan los pesos escogidos.

TRANSFER LEARNING

A menudo el problema es que se quiere entrenar una red para que clasifique un objeto determinado pero la base de datos de la que se dispone no es suficientemente grande como para obtener unos buenos resultados. En este caso se puede usar la técnica de **Transfer Learning**, que consiste en usar una red que ya ha sido entrenada previamente con otra base de datos, aprovecharse del conocimiento que ya tiene aprendido y

adaptarla para el nuevo caso. Para realizar esta adaptación se reaprovecha la información que la red ya conoce, pero se ajustan los pesos de las últimas capas. El método consiste concretamente en “congelar” las primeras capas para que no varíen, ya que contienen información genérica que puede ser reaprovechada, y volver a entrenar la red con la nueva base de datos para ajustar los pesos de las últimas capas que no estarán congeladas.

FINE-TUNING

Otro método muy extendido usado en el entrenamiento es el **Fine-tuning**, en este caso también aprovechamos el conocimiento previamente aprendido por una red, no empezamos el entrenamiento de cero, pero no necesariamente significa que transferimos el conocimiento de un entrenamiento con otra base de datos, sino simplemente que retomamos el entrenamiento de una red que ya tiene un previo conocimiento. Muy a menudo se intercambian los términos o se usan de manera indistinta, pero el método de transfer learning tiene una connotación más restrictiva que el fine tuning.

EVALUACIÓN DE LA RED

Para evaluar el comportamiento de la red una vez entrenada, es decir, para valorar si ha aprendido lo que se esperaba, se deberá reservar en el momento de preparar los datos un conjunto para usarlo en la validación. Este conjunto debe ser representativo de los datos que hay en la base de datos y suficientemente grande como para que su evaluación valga.

Una vez tenemos la red entrenada ponemos en la entrada el conjunto de datos de validación y comprobamos si a la salida obtenemos lo que esperábamos, clasificaremos las salidas usando la siguiente tabla:

		Predicción	
		Positiva	Negativa
Valor esperado	Positivo	Verdadero Positivo (VP)	Falso Negativo (FN)
	Negativo	Falso Positivo (FP)	Verdadero Negativo (VN)

Tabla 2: Evaluación de una red neuronal

A partir de esta clasificación y usando la siguiente fórmula podremos calcular la accuracy:

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}$$

En algunos casos las clases están muy poco balanceadas, es decir, el número de muestras de las que disponemos de cada clase es muy dispar y eso hace que la accuracy no nos dé un resultado que podamos considerar realista.

En base a la misma clasificación podemos obtener también las magnitudes de precision and recall:

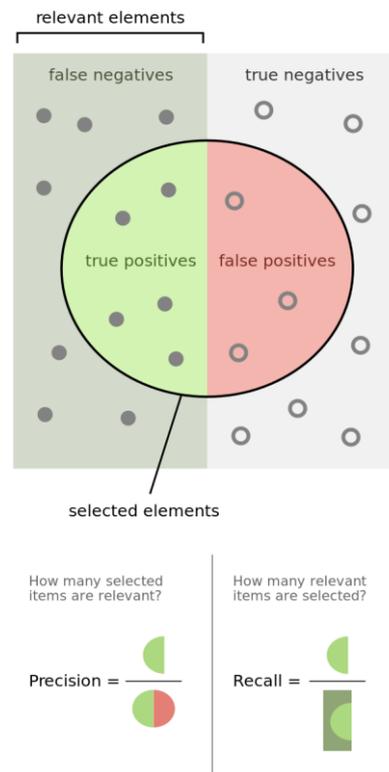


Figura 11: Precision y recall

3.3.2. R-CNN

Una red R-CNN es una red de detección que utiliza algoritmos de búsqueda selectiva para encontrar ROIs (Region of Interest) y los pasa a una CNN. Busca las áreas que podrían ser un objeto a detectar combinando píxeles similares y texturas en varias cajas rectangulares. En el artículo que se describe esta red [6] se utilizan 2000 áreas propuestas que se obtienen de la búsqueda selectiva, estas 2000 áreas se pasan por un modelo pre-entrenado de red convolucional y las salidas de esta red se pasan por un SVM [7] para ser clasificados y por una capa de regresión para determinar la posición de las bounding boxes.

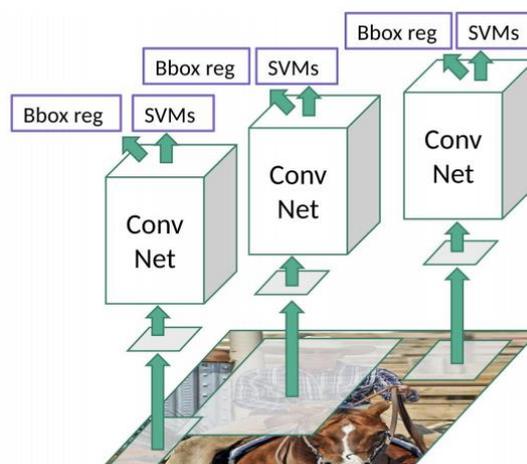


Figura 12: Arquitectura R-CNN

- **Algoritmo de búsqueda selectiva:** El procedimiento de búsqueda selectiva consiste en:
 1. Genera una sub-segmentación inicial que da lugar a varias regiones candidatas.
 2. Usando el algoritmo greedy (algoritmo voraz) se combina de forma recursiva regiones similares en regiones más grandes. El algoritmo voraz es una estrategia de búsqueda que consiste en elegir la opción óptima en cada paso local para llegar a una solución general óptima.
 3. Con las regiones generadas usando este algoritmo se obtienen los candidatos finales.

3.3.3. FAST R-CNN

En este caso, en lugar de aplicar una CNN 2000 veces a las áreas propuestas se pasa la imagen original por una CNN pre entrenada y al mapa de características de la salida de esta red se le aplican algoritmos de búsqueda selectiva. A continuación, se usa una capa de ROI pooling para mantener la medida standard y predefinida de la salida y estas salidas se pasan por una capa fully connected. Por último, se realiza la clasificación final

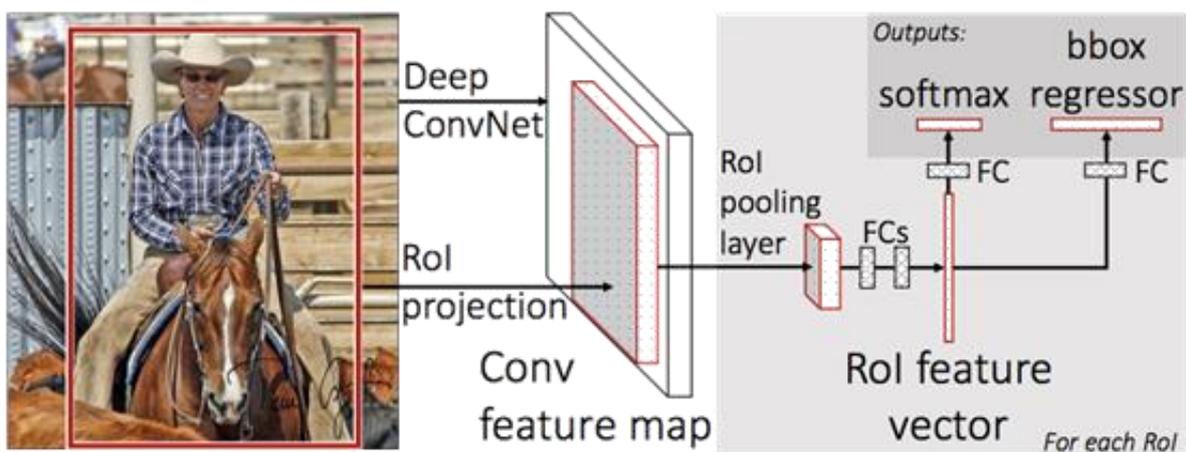


Figura 13: Arquitectura Fast R-CNN

con un clasificador softmax y se adaptan las ubicaciones de las bounding boxes con un regresor lineal.

3.3.4. FASTER R-CNN

Por último, la **Faster R-CNN** va un poco más allá y cambia los algoritmos de búsqueda selectiva por una **Region Proposal Network (RPN)**, esta red propone las regiones como lo hacían los algoritmos de búsqueda selectiva, pero de forma más precisa. La RPN está conectada a una **Fast R-CNN** que como se ha descrito en el punto anterior contiene una capa convolucional con filtros de 3x3 la salida de los cuales estará conectada a dos capas convolucionales de 1x1, una de clasificación y otra de regresión, que nos permitirán determinar si la caja propuesta es o no un objeto (clasificación) y colocar la caja correctamente (regresión). La novedad más relevante de esta red es que la RPN y la Fast R-CNN comparten convoluciones lo que permite reducir muchísimo el coste computacional.

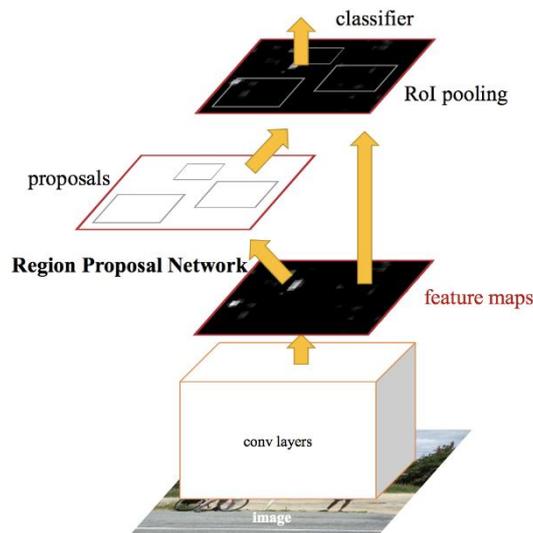


Figura 14: Arquitectura Faster R-CNN

- Region Proposal Networks (RPN):** esta red, como su nombre indica, propone regiones, a partir de una imagen en la entrada generan a la salida una serie de rectángulos que indican la ubicación donde podría encontrarse algún objeto, cada uno con un índice de "objectness", que mide la pertenencia a un conjunto de clases de objetos vs la pertenencia al fondo.

El método para generar estas propuestas consiste en deslizar una pequeña red sobre el mapa de características obtenido a la salida de la última capa convolucional compartida. Esta red es fully-connected a una ventana espacial de $n \times n$ de la entrada del mapa de características. Cada ventana deslizante se mapea en un vector de dimensión menor que se introduce en dos capas fully-connected muy similares, una capa de regression y otra de clasificación.

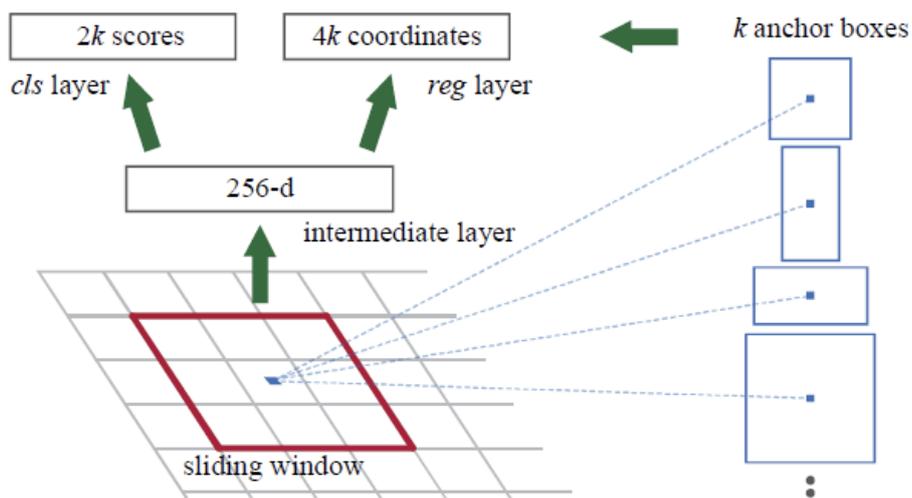


Figura 15: Salida y funcionamiento RPN

En cada posición en la que colocamos la ventana deslizante predecimos simultáneamente k regiones, de manera que la capa de regresión tendrá $4k$ elementos a la salida, que serán las 4 coordenadas de las k cajas propuestas y la capa de clasificación tendrá $2k$ índices que estimarán la probabilidad de objeto o no objeto de cada propuesta. Las k cajas propuestas se parametrizan en relación a k cajas de referencia que llamamos **anchors**. Cada anchor está centrada en la ventana deslizante correspondiente y tiene asociadas una escala y una relación de aspecto, en general se usan 3 escalas y 3 relaciones de aspecto con lo que se suele trabajar con una $k=9$ anchors.

Para entrenar una RPN se asigna una etiqueta de clase binaria a cada anchor (objeto o no objeto), se asigna una etiqueta positiva a los anchors que tienen más IoU (Intersection over-Union) con una de las cajas del ground truth o los que tienen un IoU superior a 0.7 con cualquiera de las cajas del ground truth. La etiqueta negativa se asigna a aquellos anchors cuyo IoU es menor a 0.3 para todas las cajas del ground truth. Los anchors que no son ni positivos ni negativos no se usan para entrenar.

Teniendo en cuenta estas definiciones, el objetivo del entrenamiento en este caso es el de minimizar esta función de coste para cada imagen:

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*)$$

i : índice de un anchor en un batch

p_i : probabilidad predicha de que un anchor i sea un objeto

p_i^* : 1 si la anchor es positiva o 0 si es negativa

t_i : vector que representa las coordenadas parametrizadas de la bounding box predicha.

t_i^* : vector que representa las coordenadas parametrizadas de la bounding box del ground truth asociada a un anchor positivo.

$L_{reg}(t_i, t_i^*)$ = pérdida de regresión = $R(t_i - t_i^*)$ donde R es la función de pérdida robusta.

$L_{cls}(p_i, p_i^*)$ = pérdida de clasificación.

4. Metodología:

Los métodos utilizados en este trabajo pueden separarse en dos grandes grupos:

ANOTACIÓN DE LA BASE DE DATOS:

Dado que las imágenes que tenemos no son imágenes de células aisladas, sino que son muestras de frotis de gota fina con muchas células en cada imagen inicialmente se había planteado el diagnóstico como un problema de detección: era necesario un sistema que detectase células infectadas por Malaria. Después se decidió detectar también células no infectadas para poder determinar la **parasitemia**, es decir, el número de parásitos por campo de estudio, porque en un estudio continuado del paciente este valor nos permitiría analizar su evolución y su reacción al tratamiento. Para cumplir este objetivo era necesario entonces detectar células (para conocer el número total de células) y luego clasificarlas en infectadas o no infectadas.

Con este propósito el primer paso era anotar las imágenes que servirían para entrenar la red elegida, se tenían que marcar las células infectadas y no infectadas, para que la red fuese capaz de detectar ambas clases y pudiésemos así detectar la infección y calcular la parasitemia.

La base de datos de la que disponemos está formada por 1753 imágenes distribuidas como se muestra en la tabla siguiente:

NEGATIVAS	360
Eritrocitos / otros	31
Artefactos	152
Leucocitos	89
Plaquetas	88
P.FALCIPARUM	886
Gametocitos	26
Trofozoitos	852
Esquizontes	2
No clasificadas	6
P. MALARIE	87
Gametocitos	8
Trofozoitos	23
Esquizontes	56
No clasificadas	0

P.OVALE	202
Gametocitos	23
Trofozoitos	33
Esquizontes	117
No clasificadas	29
P.VIVAX	218
Gametocitos	24
Trofozoitos	194
Esquizontes	0
No clasificadas	0
	1753

Tabla 3: Distribución base de datos

Con el fin de agilizar el proceso de anotación se valoraron distintas opciones:

- **CellProfiler [3]:** es una aplicación desarrollada por el equipo del Carpenter Lab en el Broad Institute del Harvard y el MIT. La aplicación permite hacer análisis de imágenes biológicas de distintas formas, incluyendo etiquetaje y conteo de células. Hice algunas pruebas, pero los formatos de salida de los datos no eran los que se necesitaban por lo que se terminó descartando esta opción.
- **Image Labeler Matlab [4]:** Matlab tiene una aplicación que permite marcar de manera sencilla ROIs, regiones de interés en la imagen y luego exportarlas en formato ground truth. Esta aplicación tiene implementados algunos algoritmos de etiquetado automático y además admite la posibilidad de crear algoritmos automáticos personalizados. Los algoritmos que incluye son para detectar caras o coches, por lo que no encajaban con el contexto e intenté diseñar mi propio algoritmo, pero sin éxito.
- **Aplicación de Matlab:** Descartadas las otras dos opciones se probó diseñar una aplicación propia usando App Designer de Matlab [5], la interfície permite diseñar aplicaciones de forma sencilla y usando algunos conocimientos de procesamiento de imagen se diseñó una aplicación que hace un etiquetado preliminar de las imágenes que puede ser modificado y que agiliza el proceso. El problema era que la modificación del ground truth era muy lenta porque la aplicación era poco óptima y acababa tardando más tiempo que etiquetándolas desde cero.

Finalmente, se volvió a recurrir al Image Labeler de Matlab, esta aplicación también permite importar imágenes y sus ground truth pre calculados. Usando técnicas de procesamiento de imagen se calcula una primera aproximación del etiquetado de cada imagen y se importa a la aplicación donde solo se tienen que modificar las etiquetas erróneas para conseguir el ground truth de toda la secuencia de imágenes. Este ground truth se exporta en un fichero en formato txt que es el fichero de anotaciones que se utiliza para entrenar la red.

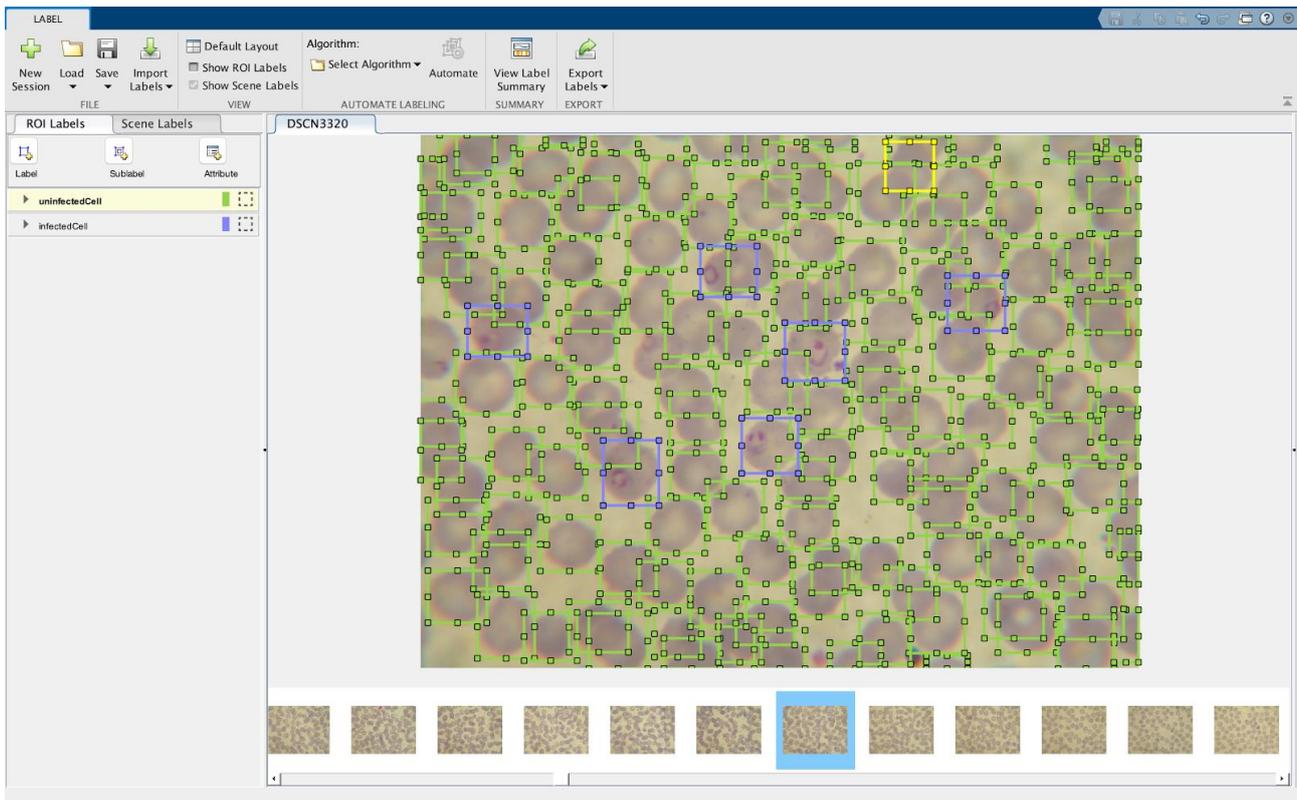


Figura 16: Image Labeler Matlab

En la figura 16 se puede ver la aplicación utilizada para el etiquetado, en morado se etiquetan las células infectadas, en verde las no infectadas y la amarilla que se puede ver arriba era la célula que estaba seleccionada en ese momento.

El procedimiento es el siguiente:

1. Usando el editor de Matlab se crea una función que permite calcular el watershed de una serie de imágenes y a partir del watershed generar una primera aproximación del ground truth basada solo en técnicas de procesamiento automático de imágenes.
Para el cálculo del watershed en Matlab me he basado en el ejemplo descrito en el siguiente enlace: <https://es.mathworks.com/help/images/marker-controlled-watershed-segmentation.html>. El procedimiento consiste en hacer una segmentación inicial que en este caso se ha realizado utilizando el gradiente y a partir de esta segmentación se generan dos conjuntos de marcadores, uno para marcar los elementos del fondo y otro para marcar los del background, estos marcadores se generan usando operadores morfológicos como los mencionados anteriormente, para conseguir que los marcadores queden siempre dentro del objeto a detectar.
2. El ground truth generado se importa en el Image Labeler, que al pasarle un objeto ground truth ya es capaz de colocar las etiquetas encima de cada imagen y utilizando esta aplicación se modifica el ground truth para arreglar aquellos etiquetajes incorrectos realizados por el algoritmo automático y para etiquetar las células infectadas que la función de Matlab no es capaz de detectar.
3. Una vez modificado el ground truth en la aplicación este puede ser exportado de nuevo al editor de Matlab, des de ahí se le da el formato necesario y se genera el fichero de anotaciones que se usará para entrenar la red.

Matlab guarda las características del ground truth de una imagen en un objeto de tipo [groundTruth](#) y las coordenadas de cada etiqueta (labelData) . Cuando lo extraemos del Image Labeler, este objeto contiene información sobre la imagen que estamos etiquetando, sobre las etiquetas y sobre donde están estas etiquetas en la imagen. (DataSource, LabelDefinitions, LabelData). Lo que se hace en el último paso por el editor de Matlab es:

1. Se convierte el ground truth a tabla, se transfiere la información que contiene el objeto ground truth a una tabla con la que se puede trabajar de manera más sencilla, para ello se ha creado en Matlab la función **gTruthToTable**.
2. Una vez tenemos la información de las imágenes, las etiquetas y de donde están colocadas estas etiquetas en cada imagen almacenada en una tabla usamos la función writetable para crear el fichero de anotaciones: en el caso de la Faster R-CNN el fichero debe ser de tipo .txt y las líneas deben tener el siguiente formato: *nombre_fichero,x1,y1,x2,y2,nombre_clase* .
3. El fichero obtenido ya puede ser utilizado para entrenar la red.

DETECCIÓN DE MALARIA:

La parte de la detección se realiza mediante una red Faster R-CNN, esta red, como se ha explicado en el apartado de marco teórico, está formada por una RPN que comparte convoluciones con una Fast R-CNN. El origen del código utilizado se puede encontrar en este github: <https://github.com/you359/Keras-FasterRCNN>, aunque en algún caso ha sido ligeramente modificado para adaptarlo a las versiones de Python, tensorflow y keras o por alguna necesidad concreta del trabajo, en este [github](#) se encuentra utilizado en el proyecto.

La red, una vez entrenada, detectará células infectadas y no infectadas en imágenes de frotis de gota fina, lo que permitirá determinar si el paciente cuya sangre estamos analizando está o no infectado de Malaria.

5. Resultados

El objetivo final del trabajo era entrenar una red para poder detectar parásitos de Malaria en imágenes de microscopio de forma automática. Tras un análisis exhaustivo de las distintas redes que estaban siendo más utilizadas hoy en día para detección e incluso de probar con alguna red que no terminó funcionando se escogió utilizar la Faster R-CNN, esta red tiene el valor añadido de que incluye una red de propuesta de regiones que la hace mucho más eficiente y rápida en la detección.

A continuación, se describen los distintos experimentos realizados y se muestran los resultados obtenidos, en cada entrenamiento se ha hecho una elección aleatoria de un número de imágenes que se han reservado para la validación del entrenamiento:

Entrenamiento de la red

Primero se entrenó la red utilizando solamente el ground truth de imágenes de frotis de gota fina que se habían etiquetado usando el método explicado en el punto anterior. Para mejorar el rendimiento de la red, se entrenó con una base de datos mucho más grande, la PASCAL_VOC, que contiene 18.000 imágenes con objetos de 21 clases diferentes para que el sistema aprendiese y mejorase sobre todo la parte de propuesta de regiones.

Utilizando los pesos obtenidos con la base de datos PASCAL VOC se volvió a entrenar el sistema con las imágenes de Malaria y se hicieron dos pruebas, en una se reentrenaba toda la red y en cambio en la otra, se mantenía la red de propuesta de regiones sin entrenar y se entrenaba solo la red de clasificación. La diferencia entre los resultados de estas dos pruebas es mínima, ya que como se verá más adelante los resultados obtenidos en la parte de clasificación de la red no son representativos ya que la parte de detección tiene un comportamiento bastante peor del deseado.

En el párrafo siguiente se explican y analizan los resultados obtenidos:

Valor de los parámetros utilizados en el entrenamiento:

Número de imágenes de entrenamiento	203
Número de imágenes de validación	48
Número de células infectadas en el subconjunto de datos de entrenamiento	357
Número de células no infectadas en el subconjunto de datos de entrenamiento	23978
Número de células infectadas en el subconjunto de datos de validación	49
Número de células no infectadas en el subconjunto de datos de validación	4486
Batch size	13

Tabla 4: Valor de los parámetros del entrenamiento

Resultados obtenidos:

Experimento 1: Entrenamiento con los pesos iniciados de forma aleatoria y solamente con las imágenes de Malaria:

Número de células detectadas en el entrenamiento	22
Porcentaje de células detectadas en el entrenamiento respecto al número total de células	0,089%
Número de células detectadas en la validación	0
Número de células infectadas detectadas en la validación	0
Número de células no infectadas detectadas en la validación	0
Porcentaje de células detectadas en la validación respecto al número total de células	0%

Tabla 5: Resultados experimento 1

Experimento 2: Entrenamiento partiendo de los pesos obtenidos de entrenar la red usando la base de datos Pascal VOC:

Número de células detectadas en el entrenamiento	490
Porcentaje de células detectadas en el entrenamiento respecto al número total de células	2,0135%
Número de células detectadas en la validación	11
Número de células infectadas detectadas en la validación	0
Número de células no infectadas detectadas en la validación	11
Porcentaje de células detectadas en la validación respecto al número total de células	0,24%

Tabla 6: Resultados del experimento 2

El análisis del comportamiento de la red no tiene sentido si se toma toda la red como única, ya que a su vez está formada por una red de detección y otra de clasificación y el comportamiento de la primera condiciona el de la segunda.

El principal problema de la red escogida se encuentra en la parte de detección de las células y es debido a la falta de imágenes y a la gran diversidad de características de las presentes en la base de datos: hay mucha variación de colores, contrastes y sobre todo formas de las células que, además, suelen estar solapadas, lo que complica su detección. Para poder ajustar suficientemente la red para obtener una detección adecuada necesitaríamos muchas más imágenes de ground truth, de las que no disponíamos.

El mal resultado del entrenamiento de la parte de detección reduce mucho el número de muestras que pasan a la parte de clasificación y esto hace que los resultados de esta

segunda parte no se puedan considerar representativos y que el cálculo de métricas como la **Precision** o **Recall** que se proponen en la parte teórica no tengan sentido.

Aun así, se ha podido comprobar que el fine-tuning mejora el resultado del entrenamiento, ya que pasamos de detectar un 0,09% de las células a detectar un 2%.

6. Presupuesto

Dado que el objetivo del proyecto no era en ningún caso fabricar un prototipo no se han considerado costes por materiales.

Tampoco se ha considerado ningún coste por el ordenador o por el servidor ya que se ha usado el ordenador del que ya se disponía y el servidor calcula, proporcionado por el grupo de Image Processing de la UPC.

Teniendo en cuenta los puntos anteriores los conceptos que generarán coste tendrán a ver con las personas que han trabajado en el proyecto y el software utilizado.

El equipo de trabajo está formado por mí (ingeniera junior) y mis dos tutoras (ingenieras senior), en cuanto al software se ha utilizado Matlab para la parte de etiquetaje y el entorno de desarrollo Pycharm para la programación de la red neuronal.

He considerado un salario de 9€ la hora para mí y 30€ la hora para mis coordinadoras; teniendo en cuenta que tuve que alargar el trabajo finalmente ha tenido una duración de 48 semanas, lo que ha comportado el coste que se muestra en la tabla siguiente:

PERSONAL		€/h	DEDICACIÓN (h/semana)	Nº semanas	TOTAL	
Ingeniera junior	1	9	30	48	12.960€	
Ingenieras senior - Coordinación	2	30	2	48	2.880€	15.840€
SOFTWARE						
Matlab	1				500€	
Pycharm	1				180€	680€
TOTAL:					16.520€	

Tabla 7: Tabla de costes

7. Conclusiones y futuros desarrollos:

Si se analiza el cumplimiento de los objetivos propuestos al inicio del trabajo se puede comprobar que en general, con algunos matices, se han cumplido:

1. He estudiado, entendido y manipulado redes neuronales. Partiendo de cero he trabajado con dos redes distintas, primero la Retinanet, que considerábamos que sería la que tendría unos mejores resultados dada la naturaleza poco balanceada de los datos, pero que tuvimos que descartar por problemas de compatibilidad con el servidor y después la Faster R-CNN que se ha acabado utilizando.
2. En la búsqueda de información sobre el estado del arte en detección de Malaria nos dimos cuenta de que muchos de los trabajos parten de células aisladas y desarrollan una red de clasificación que entrenan usando bases de datos formadas por imágenes de células recortadas, lo que elimina el problema de detección de células que es el que más empeora nuestros resultados.
3. En referencia al tercer, cuarto y quinto objetivo, se puede considerar que el método encontrado optimizaría mucho la detección de Malaria, ya que, se podría adaptar para que diagnosticase Malaria y en la misma imagen determinase la especie de parásito. Además, la mejora en el proceso de etiquetado también es bastante importante ya que permite preparar la muestra de forma rápida para poder reentrenar la red. El problema en este caso ha sido la red escogida y el número de imágenes disponibles para el entrenamiento. En futuros desarrollos podría plantearse el uso del mismo método, pero cambiando la red y sobre todo con más imágenes en el entrenamiento.

Comparar los resultados de este trabajo con los que habíamos encontrado en los artículos estudiados en el estado del arte [1][2] es complicado porque en nuestro caso la parte de detección condiciona mucho la de clasificación y el resultado obtenido a la salida de la red no se puede considerar representativo.

Haciendo autocrítica, creo que hemos hecho un buen trabajo pero que perdimos demasiado tiempo intentando desarrollar un sistema de etiquetaje automático y si se hubiera empezado antes a etiquetar ahora tendríamos más imágenes para usar en el entrenamiento y hubiéramos podido dedicar más tiempo a retocar la red.

Este trabajo puede marcar el inicio de un proyecto mucho mayor que puede culminar con el desarrollo de una aplicación móvil que permita capturar la imagen, pasarla por la red y que automáticamente detecte si el paciente tiene o no Malaria, y a la vez que esta imagen luego se utilice para aumentar la base de datos que se podrá usar si se quiere reentrenar el sistema.

Bibliografía:

Manuales:

- a. Shoklo Malaria Research Unit, International Rescue Committee BURMA BORDER PROGRAM. "**Laboratory Manual for Laboratory Technician Training**". 2002. Mae Hong Son – Mae-Sod, Tailandia
- b. Organización Mundial de la Salud. "**Medios auxiliares para el diagnóstico de las infecciones palúdicas, Segunda Edición**". Ginebra
- c. Organización Mundial de la Salud. "**Manual of basic techniques for a health laboratory, Second Edition**". Ginebra

[1] Lin, Tsung-Yi, et al. "**Focal loss for dense object detection.**" *Proceedings of the IEEE international conference on computer vision*. 2017.

[2] Pan, W. David, Yuhang Dong, and Dongsheng Wu. "**Classification of malaria-infected cells using deep convolutional neural networks.**" *Machine Learning: Advanced Techniques and Emerging Applications* 159 (2018).

[3] Cell Profiler: <https://cellprofiler.org/>

[4] Image Labeler MatLab: <https://es.mathworks.com/help/vision/ref/imagelabeler-app.html>

[5] App Designer Matlab: <https://es.mathworks.com/products/matlab/app-designer.html>

- <https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e>
- <https://towardsdatascience.com/faster-r-cnn-object-detection-implemented-by-keras-for-custom-data-from-googles-open-images-125f62b9141a>
- <https://towardsdatascience.com/review-faster-r-cnn-object-detection-f5685cb30202>
- <https://github.com/you359/Keras-FasterRCNN>

Anexos:

PLAN DE TRABAJO

El workplan ha variado respecto al que habíamos planteado originalmente por dos motivos: el primero es el cambio en los objetivos del trabajo que he comentado anteriormente, el otro motivo no tiene que ver con el temario sino con que no he podido cumplir los tiempos que había previsto lo que me ha llevado a alargar el trabajo hasta enero y por consiguiente extender los tiempos destinados a cada WP. A continuación, se describen las tareas y los hitos actualizados y en el anexo se adjunta el Gantt actualizado del proyecto:

WP1: Documento de *propuesta de proyecto y plan de trabajo*

- T1: Reuniones iniciales para discutir el alcance del proyecto.
- T2: Planificación del proyecto y redacción del documento.
- T3: Revisión del documento

WP2: Investigación sobre el estado del arte.

- T1: Diagnóstico de Malaria
- T2: Deep Learning y redes neuronales convolucionales en el procesado de imagen.
- T3: Deep learning aplicado al diagnóstico de Malaria.
- T4: Reunión con Allison (Doctorante en Castelldefels)

WP3: Investigación y aprendizaje de los temas necesarios.

- T1: Curso de Stanford (cs231n) *Convolutional Neural Networks for Visual Recognition*
- T2: Reunión en el hospital Vall d'Hebron con Francesc Zarzuela, médico investigador experto en la materia.

WP4: Red neuronal para detectar Malaria utilizando frotis de gota fina.

- T1: Selección de la red pre-entrenada a utilizar.
- T2: Selección del sistema de etiquetaje y etiquetaje de las imágenes de la base de datos.
- T3: Re-entrenamiento de la red.
- T4: Probar el funcionamiento de la red utilizando los datos de validación.

WP5: Revisión crítica

- T1: Comparar el trabajo planificado con el trabajo realizado.
- T2: Evaluar los resultados obtenidos hasta el momento y la situación en cuanto a tiempos. Escribir el documento explicando la situación y los cambios.
- T3: Revisión del documento.

WP6: Test i evaluación de los resultados.

- T1: Aprender como evaluar los resultados.
- T2: Ajustar los resultados y buscar posibles mejoras.
- T3: Implementación de las pruebas.

WP7: Memoria final

- T1: Escribir el documento.
- T2: Revisar el documento.

WP8: Presentación oral

- T1: Preparación del borrador de la presentación.
- T2: Revisión del borrador.
- T3: Ensayo de la presentación oral.
- T4: Preparación la presentación final

HITOS:

WP	Tarea	Título	Hito/ entregable	Semana
1	1	Reuniones iniciales	Documento reuniones (Drive)	1
1	2	Planificación del proyecto y escritura del documento	Borrador del documento	2,3
1	3	Revisión del documento.	ProposalWorkPlanBVG.pdf	3
2	1	Investigación sobre diagnóstico de Malaria.		1
2	2	Investigación sobre Deep Learning y CNN en procesado de imagen.		1,2
2	3	Investigación sobre Deep Learning en el diagnóstico de Malaria.		1-3
2	4	Reunión con Allison (PhD).		2
3	1	Curso cs231n Universidad de Stanford.		1-5
3	2	Reunión especialista Vall d'Hebron.		3
4	1	Selección de la red pre-entrenada a utilizar.		7-9,37-39
4	2	Etiquetaje.		9-40
4	3	Re-entrenamiento de la red.		40-47
4	4	Validación.		46-47
5	1	Modificaciones en la planificación.	Draft	8
5	2	Evaluación de los resultados.		8
5	3	Creación y revisión del documento.	CriticalReviewBVG.pdf	8
6	1	Aprender metodos de validación.		45

6	2	Ajustar resultados y buscar mejoras.		45-47
6	3	Validación.		46-47
7	1	Escribir la memoria final.	Borrador	40-47
7	2	Revisión del documento.	FinalReportBVG.pdf	46-47
8	1	Borrador presentación.	Borrador	48
8	2	Revisión de la presentación.		48
8	3	Ensayo de la presentación.		48
8	4	Presentación final.		49

