UCAB Universidad Católica ANDRES BELLO

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA INFORMÁTICA

SISTEMA INFORMÁTICO MÓVIL INTELIGENTE PARA LA DETECCIÓN DE RETINOBLASTOMA EN INFANTES

Este Jurado; una vez realizado el examen del presente trabajo ha evaluado su contenido con el resultado: Diecinueve (19).

JURADO EXAMINADOR

Firma:	Firma:	Firma:
Nombre: Marcos Salazar	Nombre: Wilmer Pereira	Nombre: José Luis Bais

REALIZADO POR MELANIE GODOY

PROFESOR GUÍA MARCOS SALAZAR

FECHA JULIO 2017

BN: 184915
AAT4616

UCAB Universidad Católica ANDRES BELLO

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA INFORMÁTICA

SISTEMA INFORMÁTICO MÓVIL INTELIGENTE PARA LA DETECCIÓN DE RETINOBLASTOMA EN INFANTES

TRABAJO ESPECIAL DE GRADO

Presentado ante la

UNIVERSIDAD CATÓLICA ANDRÉS BELLO

Como parte de los requisitos para optar al título de

INGENIERO EN INFORMÁTICA

REALIZADO POR MELANIE GODOY

PROFESOR GUÍA MARCOS SALAZAR

FECHA JULIO 2017

AGRADECIMIENTOS

Primeramente agradezco a Dios, por darme vida para lograr esta meta.

A mis padres, por apoyarme a lo largo de este camino, que no fue fácil pero nunca dudaron de mí y me apoyaron e impulsaron a seguir adelante.

A mis familiares y amigos quienes me apoyaron durante el transcurso de la carrera. Especialmente a mis amigos de la UCAB por los buenos momentos compartidos.

A mi novio, por ser un apoyo en cada paso de la carrera y durante el desarrollo de este TEG. Por creer siempre en mí y darme ánimos para continuar.

A la Doctora Livia Romero, a quien admiro, y que sin su ayuda no hubiese sido posible el desarrollo de éste Trabajo de Grado; agradezco su colaboración, interés y apoyo.

Al equipo de la Unidad de Oncología Ocular del Instituto de Oncología "Dr. Luis Razetti", quienes me recibieron con los brazos abiertos y me permitieron estar en su día a día conociendo cada caso para así poder lograr el objetivo de este proyecto.

Finalmente agradezco a todas las personas quienes directa e indirectamente ayudaron en la realización de este proyecto.

¡Muchas gracias!

DEDICATORIA

A Melissa, porque madre solo hay una y tú eres la mejor.

TABLA DE CONTENIDO

AGRADECIMIENTOS iii

DEDICATORIA iv

TABLA DE CONTENIDO v

ÍNDICE DE FIGURAS viii

SINOPSIS x

INTRODUCCIÓN 1

CAPÍTULO I 3

EL PROBLEMA 3

I.1. Planteamiento del problema 3

I.2. Objetivos 6

I.2.1 Objetivo general 6

I.2.2 Objetivos específicos 6

I.3. Alcance y Limitaciones 7

I.3.1. Alcance 7

I.3.2. Limitaciones 7

I.4. Justificación 8

CAPÍTULO II 10

MARCO REFERENCIAL 10

II.1. Cáncer infantil 10

II.2. Retinoblastoma 11

II.3. Redes neuronales 13

II.3.1. Arquitectura 13

II.4. Red Neuronal Convolucional 15

II.4.1. Estructura 16

II.5. Android SDK 18

II.6. Java 18

II.7. OpenCV 19

II.8. Clasificación de cascada 20

II.9. Tensorflow 21

II.10. Inception-v3 21

II.11. SQLite 22

CAPÍTULO III 23

MARCO METODOLÓGICO 23

III.1. Metodología Programación Extrema 23

III.1.1. Fase de Exploración 23

III.1.2. Fase de Planificación 24

III.1.3. Fase de Iteración 24

III.1.4. Fase de Producción 25

III.1.5. Fase de Mantenimiento 25

III.1.6. Fase de Muerte del proyecto 25

CAPÍTULO IV 26

DESARROLLO 26

IV.1. Fase de Exploración 26

IV.2. Fase de Planificación 27

IV.3. Fase de Iteración 28

IV.3.1. Iteración 1. Módulo de detección de rostros, ojos, iris y pupila. 28

IV.3.2. Iteración 2. Módulo de entrenamiento de la red neuronal a utilizar. 32

IV.3.3. Iteración 3. Módulo de implementación de la red neuronal entrenada. ... 36

IV.3.4. Iteración 4. Módulo de desarrollo del modelo de datos. 39

CAPÍTULO V 42

RESULTADOS 42

CAPÍTULO VI 46

CONCLUSIONES Y RECOMENDACIONES 46

VI.1. CONCLUSIONES 46

VI.2. RECOMENDACIONES 48

GLOSARIO 49

BIBLIOGRAFÍA 53

APÉNDICES 61

Apéndice 1. Historias de usuario 61

Apéndice 2. Plan de entregas 61

ÍNDICE DE FIGURAS

Figura 1. Niño con retinoblastoma en ojo izquierdo, con el síntoma de ojo blanco 12

Figura 2. Red neuronal multicapa 14

Figura 3. Capa convolucional 16

Figura 4. Operación de max pooling 17

Figura 5. Imagen original, a escala de grises y con ecualización del histograma 29

Figura 6. Región de interés delimitada para la identificación de ojos. 31

Figura 7. RDI del ojo, se busca el punto más oscuro y detecta la pupila. .. 31

Figura 8. Detección de rostro, ojos, iris y pupila en una imagen de prueba. 32

Figura 9. Resultado que arroja la red neuronal cuando se valida una imagen incluida dentro de la clasificación 'normal' 35

Figura 10. Imagen Original a la cual se le hizo la detección 38

Figura 11. Resultado arrojado en el análisis para el ojo izquierdo 38

Figura 12. Resultado arrojado en el análisis para el ojo derecho 39

Figura 13. Modelo relacional para el control de las estadísticas de la aplicación 40

Figura 14. Control de estadísticas de los análisis efectuados por la aplicación 41

SINOPSIS

El presente Trabajo Especial de Grado Titulado "Sistema Informático Móvil Inteligente Para La Detección De Retinoblastoma En Infantes", pretende mediante el uso de un algoritmo de redes neuronales y de procesamiento de imágenes, determinar si un infante posee en alguno de sus ojos retinoblastoma; identificando un patrón característico de coloración blanca en la pupila del ojo. Para esta investigación se usaron imágenes de los pacientes de la Unidad de Oncología Ocular ubicada en el Instituto de Oncología "Luis Razetti" en Caracas.

Para marcar un orden en el desarrollo de proyecto, se utilizó la Metodología Ágil *eXtreme Programming* (XP); esta metodología se rige por 6 (seis) fases, las cuales son exploración, planificación, iteraciones, producción, mantenimiento y muerte del proyecto; siguiendo cada una de estas se logró la realización exitosa del presente Trabajo Especial de Grado.

En el proyecto se utilizó *Android Studio* para el desarrollo e implementación de los algoritmos de *machine learning* y visión artificial (*OpenCV*); para el entrenamiento de la red neuronal se utilizó *Tensorflow*, basándose en el modelo *Inception*-v3.

INTRODUCCIÓN

El cáncer, en etapas tempranas es una enfermedad silente la cual dependiendo del tipo, no muestra algún síntoma que indique al paciente su presencia. El cáncer afecta a personas de cualquier edad, sexo o raza por igual.

El retinoblastoma es un tipo de cáncer que se desarrolla en el ojo, específicamente en la retina. Éste tipo de cáncer, afecta con mayor frecuencia a niños menores de 2 (dos) años de edad; éste cáncer puede afectar un solo ojo (unilateral) o ambos ojos (bilateral). El retinoblastoma puede ser heredado o no, en la mayoría de los casos no es heredado.

Al ser el retinoblastoma un cáncer que afecta el ojo y a los niños, es muy difícil de detectar a tiempo, a menos que existan antecedentes familiares de retinoblastoma. Ya que, un padre, madre o familiar no espera que su hijo recién nacido tenga un tumor en el ojo y, en la mayoría de los casos, por desinformación ni siquiera saben que existe éste tipo de cáncer.

Es importante informar a la población de esta enfermedad. Según datos de la Organización Mundial de la Salud anualmente se reportan a nivel mundial entre 7 (siete) y 8 (ocho) mil nuevos casos y en Venezuela entre 25 (veinticinco) a 30 (treinta) casos de retinoblastoma; de los cuales, muchos por desconocimiento, no son detectados a tiempo, por lo que el tumor

comienza a crecer, y cuando los síntomas ya son visibles, es posible que sea tarde para salvar la visión del ojo afectado.

Por tal razón, la intención es informar a la sociedad acerca de la existencia del retinoblastoma; y facilitar un software que ayude en la detección de dicha enfermedad.

La herramienta desarrollada ayudará a detectar si un niño presenta signos de padecer retinoblastoma. El signo más común del tumor se llama leucocoria, que se refiere al color blanco en la pupila cuando se toma una fotografía con *flash*, en vez de observarse el color rojo habitual.

El sistema, ofrece un diagnóstico aproximado por dos razones; la primera es que no toda leucocoria significa retinoblastoma, y la segunda es que la intención del sistema es la de alertar al usuario en caso de detectarse anomalía en el ojo, de suceder esto, se remite a un experto de la salud.

Tomando en cuenta que el sistema puede ayudar a salvar vidas, resulta innovador detectar la presencia de un síntoma del cáncer como lo es la leucocoria mediante una aplicación móvil que utilice técnicas de inteligencia artificial.

CAPÍTULO I

EL PROBLEMA

I.1. Planteamiento del problema

Escuchar la palabra cáncer, tiende a generar miedo, ya que, como es bien sabido puede causar la muerte si no se detecta a tiempo; sin embargo, según el Instituto Nacional del Cáncer de Estados Unidos, indica que en 2014 (dos mil catorce) se registraron 14,5 (catorce punto cinco) millones de personas que viven luego de ser diagnosticados con dicha enfermedad.

El cáncer se caracteriza por la reproducción anormal de células; cuando las células del cuerpo envejecen, éstas mueren y nuevas células las reemplazan. Cuando un órgano está afectado con cáncer, el proceso de reproducción de células se descontrola y, las células que deben de morir no lo hacen y se generan mutaciones que no pueden ser reparadas, y se forman masas llamadas tumores.

Existen tumores benignos y malignos. Los primeros tienen un crecimiento lento y no se propagan a otras partes del cuerpo, y los segundos son de rápido crecimiento, se pueden expandir a otros lugares del cuerpo y reproducirse luego de ser extirpados, provocando en la mayoría de los casos una muerte inexorable en un periodo de tiempo variable.

Un tumor cancerígeno se podría alojar en cualquier órgano y se puede desarrollar a cualquier edad. El origen de estas anormalidades se puede deber

a varios motivos; agentes carcinógenos como la radiación ultravioleta, productos químicos, humo de tabaco, sin embargo, uno de los principales se basa en la genética particular de cada ser humano; dichas alteraciones genéticas pueden ser heredadas o producidas en las células por un virus o una lesión provocada de manera externa.

Dentro de los distintos tipos de cáncer, está el retinoblastoma. El retinoblastoma es un tumor que se desarrolla en la retina por la mutación de la proteína Rb, este tumor se presenta mayormente en niños, constituye la primera causa de malignidad intraocular, en Estados Unidos la incidencia anual es de cuatro por cada millón de niños. El 40% (cuarenta por ciento) de los casos ocurren por herencia, tendiendo a afectar ambos ojos y afectan a niños pequeños; el otro 60% (sesenta por ciento) es por carácter esporádico, es decir, no hereditario, usualmente solo se compromete un ojo y afecta a adolescentes menores de 15 (quince) años. Este tipo de cáncer es el que se ocupará como tema central de estudio para el presente trabajo de grado.

El retinoblastoma presenta una serie de síntomas los cuales permiten detectarlo a tiempo; como estrabismo convergente, visión doble, ojos desalineados, enrojecimiento y dolor en el ojo, visión deficiente; iris de distinto color en cada ojo; y el síntoma que será motivo de estudio en el presente Trabajo de Grado es cuando en uno o ambos ojos se puede observar la pupila de color blanco o manchas blancas al tomar una fotografía con una cámara con *flash*.

En Venezuela anualmente se registran treinta casos de retinoblastoma; los cuales muchas veces por la desinformación que existe no se detectan a tiempo; afectando grandemente la vida de un paciente, puesto que los daños pueden ir desde la pérdida del ojo hasta la muerte del paciente. Al existir una gran desinformación, es posible que no se tome como alerta el síntoma objeto de este estudio, que es la coloración blanca de la pupila al tomar una fotografía con *flash*; es por esto que cada persona que tenga un dispositivo móvil con *flash* debe conocer acerca de esta enfermedad para ayudar a salvar una vida.

La cámara fotográfica es posiblemente la herramienta con más uso en un *Smartphone*; por ello, parece oportuno desarrollar un sistema informático móvil, a manera de que al tomar una fotografía de un niño en cualquier entorno utilizando un *Smartphone*, mediante el reconocimiento de la cara y del ojo del niño, se pueda realizar un diagnóstico preventivo si sufre o no de retinoblastoma, ayudando a detectar esta enfermedad a tiempo y de una manera fácil con una herramienta al alcance de todos, ya que, según CONATEL, solo en Venezuela para el año 2015 (dos mil quince), más de 11 (once) millones de personas poseía un teléfono móvil inteligente que les permita descargar la aplicación.

I.2. Objetivos

I.2.1 Objetivo general

Desarrollar un sistema informático móvil que mediante el uso de la cámara incorporada y del reconocimiento de imágenes permita el análisis en tiempo real de la pupila y determine si el ojo de un niño está afectado por Retinoblastoma.

I.2.2 Objetivos específicos

- Definir y desarrollar el método de detección de rostros de infantes desde la cámara del teléfono en tiempo real.
- Definir y desarrollar el método de detección de los ojos en el rostro del infante en tiempo real desde la cámara del teléfono.
- Definir y desarrollar el método de detección de iris y pupilas en los ojos detectados del infante en tiempo real desde la cámara del teléfono.
- Definir las características a codificar que constituyen la base de conocimientos de entrada de la red neuronal artificial a implementar, como estrategia de detección y reconocimiento de retinas afectadas por retinoblastoma.
- Desarrollar la red neuronal artificial mencionada en el punto anterior, capaz de establecer el patrón que identifique en imagen las características del Retinoblastoma.

- Diseñar y desarrollar el modelo de datos que de soporte a la persistencia de información.
- Diseñar y desarrollar el modelo de interacción hombre-máquina para que la aplicación sea intuitiva y de fácil uso.

I.3. Alcance y Limitaciones

I.3.1. Alcance

- Ayudar al usuario de la aplicación conocer si el niño capturado por la cámara, padece de cáncer de retina en el ojo, basándose en si la coloración de la pupila es blanca.
- La aplicación realizará el reconocimiento de imágenes si sólo se captura un niño al cual efectuar el reconocimiento facial y ocular.

I.3.2. Limitaciones

- Para el desarrollo de la aplicación móvil se utilizará el lenguaje de programación *Android SDK* 25.0.0 y el IDE *Android Studio* 2.3.1.
- El desarrollo de la aplicación funciona para teléfonos móviles inteligentes que tengan sistema operativo *Android* 4.0 *IceCream Sandwich* o superior y una cámara de 8 megapíxeles como mínimo con *flash* operativo.
- Se detectará solamente 1 (un) niño en la foto.

- La imagen se puede capturar a una distancia mínima de 50 (cincuenta) centímetros y máxima de 1,5 (uno punto cinco) metros.
- Para mejores resultados la imagen debe ser capturada en condiciones de luz clara, preferiblemente bajo luz blanca. La posición del rostro debe estar de frente a la cámara.
- Las imágenes analizadas serán obtenidas de fuentes certificadas, como fundaciones o asociaciones y de los pacientes de la Unidad de Oncología Ocular ubicada en el Instituto de Oncología Dr. Luis Razetti.
- Para el desarrollo de la red neuronal, se utilizará la librería *Tensorflow* de Google basándose en el modelo *Inception* v3.

I.4. Justificación

En el año 2015 (dos mil quince), aumentó en 43% (cuarenta y tres por ciento) la cantidad de personas a nivel mundial que poseen un *Smartphone*, y solamente en Venezuela poco más de 11 (once) millones de personas son dueños de uno de estos teléfonos inteligentes; dando cabida así a los miles de usos que tienen estos pequeños aparatos.

En esta era tecnológica, donde el teléfono ha sustituido a gran cantidad de dispositivos, lo más lógico es que se desee hacer de todo con él; desde registrar un número telefónico hasta poder guardar grandes cantidades de fotografías; estas últimas son muy populares entre los usuarios de *Smartphones*, la mayoría las utiliza para guardar momentos que valen la pena

recordar, pero que tal si aparte de registrar los momentos importantes de la vida de una persona, también puedan salvarle la vida a otra.

Por tal razón, se desea crear para uno de los dispositivos más utilizados en los últimos años como lo es el teléfono inteligente, una aplicación que permita la detección del Retinoblastoma en los niños; esta aplicación hará uso de la cámara trasera del *Smartphone*, la cual detectará en tiempo real el rostro, ojos y pupila y permitirá capturar la imagen de los ojos, la cual será analizada por el sistema para determinar el diagnóstico.

Los niños son por mucho, el objeto de fotografía para sus padres, ya que, estos quieren registrar el crecimiento de su hijo en su teléfono personal. El aporte de este trabajo de investigación es entregar en las manos de cualquier persona que posea un teléfono inteligente, la posibilidad de detectar el retinoblastoma y salvarle la vida a un pequeño.

CAPÍTULO II

MARCO REFERENCIAL

El presente capítulo describe los fundamentos teóricos requeridos para lograr el cumplimiento del proyecto. Los conceptos descritos serán utilizados para lograr un mejor entendimiento del proyecto y adicionalmente los programas que fueron utilizados para su desarrollo.

II.1. Cáncer infantil

El término "cáncer infantil" usualmente se utiliza para indicar distintos tipos de cáncer que pueden aparecer en los niños antes de cumplir los 15 (quince) años. El cáncer infantil es poco frecuente, las tasas mundiales de incidencia de acuerdo a la Organización Mundial de la Salud (OMS), oscilan entre 50 (cincuenta) y 200 (doscientos) por cada millón de niños en el mundo.

Las características de la enfermedad en la niñez difieren de las observadas en enfermos de otros grupos de edad. La leucemia, representa alrededor de una tercera parte de todos los cánceres infantiles. Los otros tumores malignos más comunes en niños son los linfomas y los tumores del sistema nervioso central. Existen varios tipos tumorales que se dan casi exclusivamente en los niños, como los neuroblastomas, los nefroblastomas, los meduloblastomas y los retinoblastomas.

Mayormente los cánceres infantiles presentan signos y síntomas inespecíficos, lo que puede hacer que se detecten en fases ya avanzadas. En los países con altos ingresos, donde los niños suelen estar bajo una estricta vigilancia tanto médica como parental, las posibilidades de una detección a tiempo son mucho más elevadas. En cambio, en los países con escasos recursos la falta de acceso a los servicios de salud dificulta la detección temprana de la enfermedad.

II.2. Retinoblastoma

El retinoblastoma es un cáncer de ojo que se origina en la retina; la retina es el tejido nervioso que cubre el interior de la parte posterior del ojo. Se encarga de detectar la luz y enviar la información al cerebro a través del nervio óptico. Un retinoblastoma puede ser hereditario, que representa del 25% (veinticinco por ciento) al 30% (treinta por ciento) de los casos y no hereditario que representa del 70% (setenta por ciento) al 75% (setenta y cinco por ciento) de los casos.

El retinoblastoma hereditario puede manifestarse en uno o los dos ojos, los afectados por este tipo de retinoblastoma se presume que son todos los que tienen la enfermedad bilateral y el 15% con la enfermedad unilateral. Por lo general afecta a niños pequeños, dos tercios de los casos se diagnostican antes de los de 2 (dos) años de acuerdo al Instituto Nacional del Cáncer de Estados Unidos.

El retinoblastoma se presenta cuando las células nerviosas de la retina mutan genéticamente, lo que causa que estas crezcan y se multipliquen permitiendo que se forme un tumor. Lo más común es que las células se propagan en el ojo y las estructuras cercanas, pero también se pueden extender a otras áreas del cuerpo (metástasis), incluyendo el cerebro y la columna vertebral.

Cuando un ojo está afectado por retinoblastoma, se puede observar en la pupila un color blanco cuando la luz brilla en el ojo, por ejemplo, el *flash* de una cámara fotográfica; siendo éste el síntoma más común. El retinoblastoma también tiene otros síntomas, los cuales pueden incluir ojos mirando en diferentes direcciones, enrojecimiento o hinchazón de los ojos, estrabismo.

Figura 1. Niño con retinoblastoma en ojo izquierdo, con el síntoma de ojo blanco.

Fuente: ¿Qué Es un Retinoblastoma? (Ene., 2012). American Academy of Ophthalmology. Disponible en: https://www.aao.org/salud-ocular/enfermedades/que-es-retinoblastoma.

II.3. Redes neuronales

Las redes neuronales son un tema muy importante dentro de la inteligencia artificial. Su nombre viene de la idea de imitar el comportamiento del cerebro humano, específicamente el funcionamiento de las neuronas y sus conexiones; tratando así de crear modelos artificiales que solucionen problemas difíciles de solventar mediante el uso de algoritmos convencionales.

A pesar de lo mencionado anteriormente, la referencia biológica no ha sido muy útil, por tal razón las redes neuronales han ido moviéndose para tener un foco en matemática y estadística. Se basan en la idea de, dados unos parámetros, éstos se combinan para predecir un resultado. Para saber cómo combinar estos parámetros, se procede a entrenar la red neuronal; y una vez ya entrenada la red, se puede utilizar para realizar predicciones o clasificaciones, es decir, aplicar la combinación.

II.3.1. Arquitectura

La arquitectura de una red neuronal es la organización de las neuronas en su interior y está fuertemente ligada al algoritmo de aprendizaje utilizado en el entrenamiento de la red.

Las redes neuronales de acuerdo al número de capas, se dividen en redes monocapa y redes multicapa. En las redes monocapa el primer nivel está relacionado con el vector de entrada y tiene la dimensión del mismo, la información entra al primer nivel y los datos son transferidos

siendo modificado por los pesos sinápticos; la arquitectura solo posee una capa de procesamiento, de ahí su nombre. Las redes multicapa tienen un nivel de entrada con *n* neuronas, y una capa de salida de *m* neuronas; incluye una capa intermedia que está conformada por *i* neuronas, ésta capa recibe el nombre de Capa Oculta.

La otra clasificación de las arquitecturas de redes neuronales es dependiendo de cómo fluye la información, dividiéndose en redes *feedforward* y redes recurrentes. En las redes *feedforward*, la información fluye en un solo sentido, desde la capa de entrada, luego a la o las de procesamiento, hasta llegar a la capa de salida. Por otro lado, están las redes recurrentes, en este tipo de red la información no siempre fluye en un solo sentido, ya que puede alimentarse de capas anteriores a través de conexiones sinápticas.

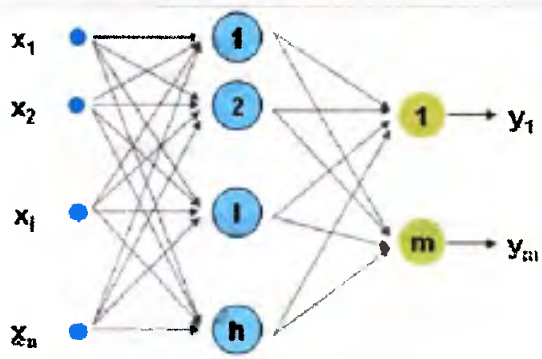

Figura 2. Red neuronal multicapa

Fuente: Redes Neuronales Artificiales Arquitecturas y Aprendizaje

Disponible en: http://www.academia.edu/8172191/Redes_Neuronales_Artificiales_Arquitecturas_y_Aprendizaje

II.4. Red Neuronal Convolucional

Es una red neuronal que está diseñada fundamentalmente para el procesamiento de datos estructurados, tales como imágenes o señales de voz. Su ventaja principal es que la cantidad de parámetros a entrenar es menor que en una red multicapa con el mismo número de capas ocultas, haciendo así que su entrenamiento sea más rápido.

Las redes neuronales convolucionales son similares a las redes neuronales artificiales comunes, están compuestas por neuronas que tienen pesos y sesgos que pueden aprender. Cada neurona recibe entradas, realiza entre ellas un producto escalar y luego aplica una función de activación.

Las redes convolucionales solucionan el problema de las redes ordinarias al escalar correctamente imágenes de alta definición. Estas funcionan formando pequeños pedazos de información, y luego combinando dicha información en las capas más bajas de la red.

Un modo sencillo de entender las redes convolucionales es por su funcionamiento, que en líneas generales empieza con la primera capa que detecta los bordes y establece patrones; las capas siguientes los combinan en formas más simples, y finalmente, en patrones de los objetos en posiciones variadas dependiendo de la iluminación, tamaño, etc. Y para culminar, las capas finales trataran hacer coincidir una imagen de entrada con los patrones y generar una predicción final.

II.4.1. Estructura

Las redes convolucionales están construidas con una estructura que contiene 3 tipos de capas.

La capa convolucional, es la que le da el nombre a la red, la diferencia respecto a cualquier otra red neuronal es el uso de una operación llamada convolución en sus capas; en vez de aplicar la usual multiplicación de matrices. El procedimiento de convolución admite como entrada la imagen, sobre ella se aplica un filtro que devuelve un mapa con las características de la imagen original, logrando de esa forma la disminución del tamaño de los parámetros. La convolución permite trabajar con entradas de varios tamaños.

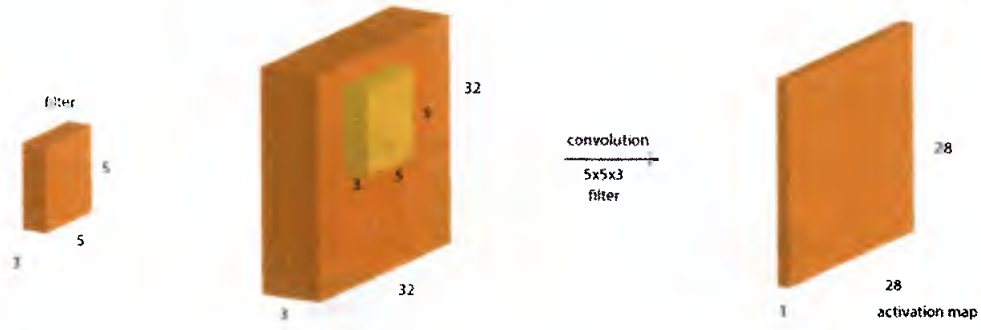

Figura 3. Capa convolucional

Fuente: Redes neuronales convolucionales con TensorFlow

Disponible en: http://relopezbriega.github.io/blog/2016/08/02/redes-neuronales-convolucionales-con-tensorflow/

La capa de reducción o *pooling*, es la que reducirá los parámetros hasta quedar con las características más comunes. Por lo general, esta capa se coloca luego de la capa convolucional. Su principal utilidad es la de reducir las dimensiones espaciales (ancho x alto) del volumen de entrada. La operación realizada por esta capa también se le llama reducción de muestreo, puesto que la reducción de tamaño conduce a pérdida de información. Aunque, la pérdida de información puede beneficiar la red, ya que reduce la carga de cálculo para las capas siguientes y reduce el sobreajuste.

La operación que es frecuentemente utilizada en esta capa se llama *max-pooling*, la cual se basa en dividir la imagen de entrada en un conjunto de rectángulos y, respecto de cada subregión, tomar el máximo valor de la misma.

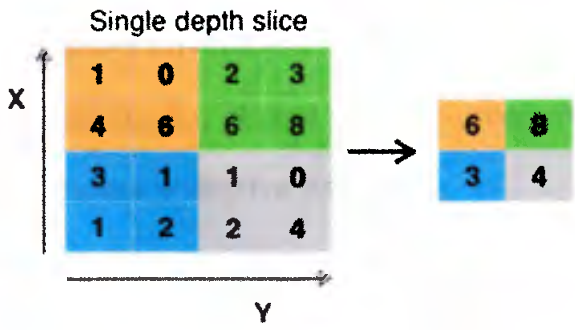

Figura 4. Operación de max pooling

Fuente: Redes neuronales convolucionales con TensorFlow

Disponible en: http://relopezbriega.github.io/blog/2016/08/02/redes-neuronales-convolucionales-con-tensorflow/

Finalmente se encuentra la capa clasificadora totalmente conectada, esta capa está al final de la capa convolucional y la capa de *pooling,* las redes utilizan normalmente capas totalmente conectadas en la que cada *pixel* se toma como una neurona separada tal como en una red neuronal ordinaria. Esta última capa clasificadora va a tener tantas neuronas como cantidad de clases a predecir.

II.5. Android SDK

Un SDK (*Software Development Kit*) o kit de desarrollo de software, es un cúmulo de herramientas que permiten programar aplicaciones para un entorno tecnológico particular. El *Android* SDK es un conglomerado de herramientas y librerías desarrolladas por *Google* que permiten desarrollar, compilar y depurar aplicaciones para el sistema operativo *Android.*

Las aplicaciones para *Android* son programadas en lenguaje Java, y el IDE (*Integrated Developmet Enviroment*) o ambiente de desarrollo integrado más utilizado es *Android Studio,* el cual permite programar de forma nativa aplicaciones para el sistema operativo *Android.*

II.6. Java

Java es un lenguaje de programación orientado a objetos, desarrollado en 1991 por *Sun Microsystems.* La intención de *Sun* era crear un lenguaje con

una estructura y sintaxis similar a C y C++, pero con un modelo de objetos más simple y eliminando las herramientas de bajo nivel.

Java está sustentado en cinco pilares, la programación orientada a objetos, la posibilidad de ejecutar un mismo programa en varios sistemas operativos, la incorporación de soporte para trabajo en red, la ejecución del código en sistemas remotos de forma segura y la facilidad de uso.

II.7. OpenCV

Open Source Computer Vision, es una librería software libre de visión artificial y *machine learning*.

Opencv tiene una licencia BSD *(Berkeley Software Distribution)* que es una licencia de software permisiva, dando la opción de modificar el código; cuenta con una comunidad de 47000 (cuarenta y siete mil) personas y más de 7 (siete) millones de descargas. La librería tiene más de 2500 (dos mil quinientos) algoritmos, entre los cuales hay algoritmos de *machine learning* y de visión artificial para utilizar.

Dichos algoritmos permiten la identificación de objetos, caras, clasificación de acciones humanas, *tracking* de movimientos de objetos, acertar entre imágenes similares, eliminar ojos rojos, seguir el movimiento de los ojos, reconocer escenarios, entre otras aplicaciones.

Tiene uso en aplicaciones tales como la detección de intrusos en vídeos, inspeccionar etiquetas en productos, etc.

OpenCV está escrito en C++, tiene interfaces en C++, C, Python, Java y MATLAB *interfaces* y funciona en Windows, Linux, Android y Mac OS.

II.8. Clasificación de cascada

El clasificador en cascada para la detección de objetos fue propuesto por Paul Viola (2001) y mejorado por Rainer Lienhart (2002).

En primer lugar se forma un clasificador con cientos de vistas de muestra de un objeto particular (como un rostro, un carro, ojos, etc.), los cuales llevan por nombre ejemplos positivos, los cuales son escalados a un mismo tamaño; y luego están los ejemplos negativos, que son imágenes arbitrarias del mismo tamaño.

Luego que el clasificador es entrenado con dichos cientos de imágenes, pueden ser aplicados a una región de interés en una imagen de entrada. El clasificador devuelve un "1" si es probable que la región muestre el objeto (cara, carro, etc.), y "0" en caso contrario. El clasificador está diseñado para que pueda ser fácilmente "redimensionado" lo que le permitirá encontrar los objetos de interés en diferentes tamaños.

La palabra cascada en el nombre del clasificador significa que el clasificador resultante consta de varios clasificadores más simples (etapas)

que son aplicados posteriormente a una región de interés hasta que en algún momento el candidato es rechazado o todas las etapas pasan.

II.9. Tensorflow

TensorFlow es una librería *open source* desarrollada por *Google* que permite efectuar cálculos numéricos mediante el uso de diagramas de flujo de datos. Los nodos del grafo simbolizan operaciones matemáticas, y los arcos del grafo simbolizan arreglos de datos multidimensionales (tensores) que se comunican entre ellos.

Tensorflow es utilizado en productos de *Google* tal como *Gmail*, *Search*, reconocimiento de voz, *Google Translate*, entre otros. El core de *Tensorflow* está desarrollado en lenguaje C++, ofrece librerías en python y C++ y puede ser utilizado sobre CPU, GPU, en sistemas operativos Linux, Mac OS, y también en plataformas móviles como *Android* e *iOS*.

II.10. Inception-v3

Inception-v3 es el nombre de un modelo de aprendizaje profundo pre-entrenado. Fue desarrollado por Google y entrenado para el Desafío de Reconocimiento Visual de *ImageNet*, utiliza los datos de 2012 (dos mil doce), y puede diferenciar entre 1000 (mil) clases diferentes.

Esta red es posible reciclarla para así distinguir un pequeño número de clases basadas en ejemplos personalizados. Al hacer eso, se realiza lo que se llama re-entrenamiento de la red, donde solo se entrena la capa final del modelo, reduciendo así los tiempos de entrenamiento.

II.11. SQLite

SQLite es una librería que implementa un motor de base de datos SQL autónomo, sin servidor, de configuración cero y transaccional. *SQLite* es un motor de base de datos *SQL* incorporado. A diferencia de otras bases de datos *SQL*, *SQLite* no tiene un proceso de servidor separado; lee y escribe directamente en archivos de disco ordinarios.

SQLite es una base de datos muy eficaz para ambientes embebidos, ya que ofrece un alto rendimiento, seguridad, estandarización e inter operatividad. Todo esto la ha convertido en la base de datos más factible para el desarrollo de aplicaciones móviles.

CAPÍTULO III

MARCO METODOLÓGICO

En este capítulo se describe la metodología utilizada para el desarrollo del proyecto. Para el desarrollo de este Trabajo Especial de Grado se implementó la metodología ágil de Programación Extrema o *eXtreme Programming* (XP); esta es una de las metodologías que más popularidad ha tomado, ya que, se basa en que es posible desarrollar software de calidad manteniéndolo simple, y permitiendo la adaptación rápida al cambio; contempla que durante el ciclo de desarrollo de un proyecto siempre se presentarán cambios y que de hecho es algo bueno.

III.1. Metodología Programación Extrema

La metodología XP es una metodología ágil para el desarrollo de software, formulada por Kent Beck (1999). Según Kent Beck en su libro *Extreme Programming Explained* (Addison-Wesley, 1999), el ciclo de vida de un proyecto ideal implementando XP se basa en las siguientes seis (6) fases.

III.1.1. Fase de Exploración

Durante la primera fase, son llevadas a cabo todas las actividades referentes al levantamiento de información para así poder definir los requerimientos. En esta fase se determinan las historias de usuario las cuales

son necesarias para la planificación del proyecto, se determinan las tecnologías a utilizar en el desarrollo del proyecto y finalmente se determina la arquitectura del sistema.

III.1.2. Fase de Planificación

Durante esta fase, se define la prioridad de cada historia de usuario dentro del sistema, estimando que historia de usuario contendrá cada iteración.

La Planificación de publicaciones (*release plan*). Una vez se encuentran definidas las historias de usuario, se debe definir qué historias de usuarios estarán en cada versión del programa; así como las fechas en la que serán publicadas dichas versiones.

III.1.3. Fase de Iteración

Dependiente del plan de iteraciones se desarrolla la arquitectura del sistema y el Plan de entregas, detallando las historias de usuario que contendrá cada iteración. En cada iteración se realiza el ciclo de análisis y planificación, diseño, implementación y pruebas.

El diseño e implementación de pruebas durante esta fase permiten que la metodología XP sea eficaz, ya que, se van aplicando sobre el código a medida que éste se va desarrollando, permitiendo encontrar errores antes

que el sistema pase a producción, minimizando el tiempo de resolución de errores.

III.1.4. Fase de Producción

Una vez son terminadas cada una de las iteraciones, se debe llevar el producto a la fase de producción, a este nivel, se tiene el sistema totalmente integrado y funcional. Se realizan todas las pruebas necesarias para garantizar el correcto funcionamiento del sistema, afinando el rendimiento del mismo.

III.1.5. Fase de Mantenimiento

El mantenimiento es el estado normal durante un desarrollo basado en la metodología XP, se debe ser muy cuidadoso al realizar cambios en el sistema durante el mantenimiento ya que las funcionalidades previas del sistema deben continuar activas durante el proceso.

III.1.6. Fase de Muerte del proyecto

Es la fase final del desarrollo, una vez culminado el mismo, se debe realizar la respectiva documentación del sistema para brindar apoyo durante su uso.

CAPÍTULO IV

DESARROLLO

El presente capítulo tiene como finalidad describir los pasos seguidos para desarrollar el presente Trabajo Especial de Grado.

Como se indicó en el capítulo anterior la metodología aplicada para desarrollo del proyecto fue la metodología XP. Surgieron un total de tres (3) iteraciones, donde cada una de las cuales cuenta con las distintas fases de la metodología.

Es indispensable destacar que las fases abordadas durante el desarrollo del proyecto serán únicamente las tres (3) primeras de la metodología XP, ya que no se llevaron a cabo el resto de ellas. A continuación, se presentan las etapas por las que transcurrió el sistema realizado en este Trabajo de Grado.

IV.1. Fase de Exploración

Esta fase tuvo una duración de aproximadamente tres semanas. Durante las cuales, se llevaron a cabo reuniones con el tutor, para determinar cuáles son los factores más importantes a considerar para efectuar correctamente el reconocimiento del retinoblastoma en infantes. También se llevó a cabo la definición del entorno de desarrollo, y familiarización con el mismo; el entorno seleccionado fue *Android Studio* 2.3.1, versión del SDK 25 y el dispositivo en que se instaló para ser presentado fue un Motorola modelo Moto X, con sistema operativo *Android* 5.1 y cámara de

10 *megapíxeles.* Posteriormente se procedió a analizar y recopilar las historias de usuario a desarrollar, las mismas pueden apreciarse en el *Apéndice 1.*

IV.2. Fase de Planificación

Luego de definir en la fase de exploración las historias de usuario a desarrollar, se creó un plan de entregas el cual contempla la prioridad de cada iteración y el orden en el cual deben ser desarrolladas las historias de usuario. Cada componente dentro del cronograma corresponde a una iteración de la metodología de desarrollo. Luego de culminar la iteración se realizan pruebas para comprobar su funcionamiento y al ser correcto se da por terminada y se pasa a la siguiente.

Dentro del plan de proyecto, el cual puede ser apreciado en el *Apéndice 2*, está el cronograma donde se estipula el tiempo estimado que tomará el desarrollo de cada iteración. El orden de las iteraciones será el siguiente:

- Módulo de detección de rostros, ojos, iris y pupila.
- Módulo de entrenamiento de la red neuronal a utilizar.
- Módulo de implementación de la red neuronal entrenada.
- Módulo de desarrollo del modelo de datos.

A continuación se especifica cada una de las iteraciones divididas por etapas dentro del marco de la metodología XP.

IV.3. Fase de Iteración

IV.3.1. Iteración 1. Módulo de detección de rostros, ojos, iris y pupila.

IV.3.1.1. Planificación

En el comienzo de esta iteración se realizó el levantamiento de información y requerimientos correspondientes al módulo de detección de rostros, ojos, iris y pupila.

Este módulo debe encargarse de detectar en tiempo real, mediante el uso de la cámara del dispositivo móvil un rostro, los ojos de dicho rostro y la pupila e iris de cada ojo; esto con el fin de obtener una imagen del ojo la cual pueda ser utilizada para determinar la presencia del síntoma del retinoblastoma llamado leucocoria.

IV.3.1.2. Diseño

Para el diseño de esta iteración, fue necesario tomar en cuenta distintos factores, tales como que la técnica a utilizar sea aplicable en dispositivos móviles con sistema operativo (S.O.) *Android*, que permita la detección en tiempo real desde la cámara del dispositivo y por supuesto, que su aplicación esté orientada al lenguaje de programación Java.

Por tal razón, se eligió OpenCV como método de detección de rostros, ojos, iris y pupila, ésta librería de código abierto brinda una extensa variedad de algoritmos aplicables a tecnologías móviles; dentro de los cuales se encuentran los de detección que requiere el sistema.

IV.3.1.3. Implementación

Para comenzar, se va de lo general a lo específico, siendo lo general la detección de rostros y lo específico la detección de pupila. Para realizar la detección de objetos, OpenCV cuenta con sus propios clasificadores entrenados para dicha tarea, éstos se almacenan en archivos de extensión XML.

Para detectar un rostro con OpenCV lo primero es procesar la imagen en la cual se desea detectar un rostro; luego se deben aplicar una serie de cambios a dicha imagen para que los algoritmos funcionen correctamente, ya que, se realizará lo que se conoce como un procesamiento digital de imágenes; el primer paso es la conversión de la imagen capturada por la cámara del dispositivo en tiempo real, a escala de grises.

Luego de que la imagen se convierte a escala de grises, se realiza al ecualización del histograma, dicha conversión estandariza el contraste y el brillo de la imagen, esto es para que aun bajo distintas condiciones de luz no se vea afectada la detección del rostro en la imagen.

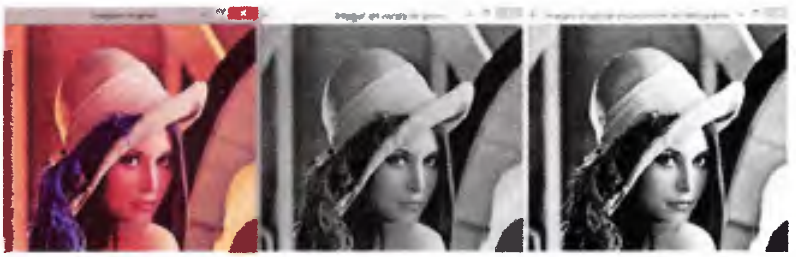

Figura 5. Imagen original, a escala de grises y con ecualización del histograma

Fuente: Elaboración propia.

Disponible en: http://docs.opencv.org/2.4/_images/lena.png

Al tener la imagen procesada, se carga el detector a utilizar, el cual lleva por nombre "lbpcascade_frontalface.xml", este clasificador está guardado dentro de la carpeta del proyecto, se carga mediante un método de carga de archivo. Este clasificador permite la detección de rostros eficazmente respecto a otros clasificadores para detectar rostros de OpenCV, y también cuenta con mejor rendimiento en dispositivos móviles.

El siguiente paso es la detección de ojos, al igual que con la detección de rostro la imagen debe sufrir una transformación para que sea más sencilla la detección mediante los algoritmos de OpenCV. Los ojos serán detectados una vez se detecte un rostro, debido a ello se utiliza la misma imagen ya convertida, y se procede a computar el área de los ojos.

Al detectar un rostro se reduce la Región De Interés (RDI) y permitirá ubicar y detectar los ojos. Para la detección de ambos ojos, se utilizó el clasificador "haarcascade_lefteye_2splits.xml", este clasificador brinda mejores resultados respecto a otros clasificadores, cuando realiza la detección de ambos ojos. Luego de detectar los ojos, se dividen en ojo izquierdo y derecho para ser analizados por separado.

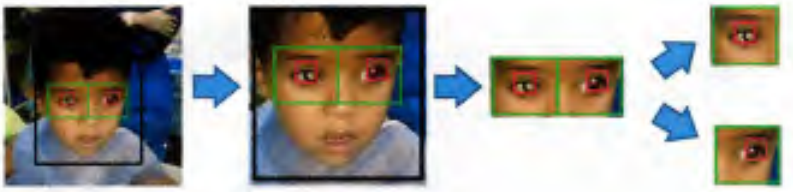

Figura 6. Región de interés delimitada para la identificación de ojos.

Fuente: Aplicación desarrollada, LeukoDetect.

Al tener los ojos detectados de manera separada, se realiza la detección de la pupila e iris. Dentro de la RDI de los ojos se realiza la búsqueda del punto más oscuro o claro, dependiendo de la imagen, dando como resultado la detección de la pupila.

Figura 7. RDI del ojo, se busca el punto más oscuro y detecta la pupila.

Fuente: Aplicación desarrollada, LeukoDetect.

IV.3.1.4. Pruebas

Luego de culminar la fase de desarrollo, se procede a realizar las pruebas, donde se obtendrá en tiempo real mediante el uso de la cámara del dispositivo móvil, la detección de rostro, ojo, iris y pupila; dando así por culminada la primera iteración. El resultado de las pruebas se encuentra reflejado en la *Figura 8*.

Figura 8. Detección de rostro, ojos, iris y pupila en una imagen de prueba.

Fuente: Aplicación desarrollada, LeukoDetect.

IV.3.2. Iteración 2. Módulo de entrenamiento de la red neuronal a utilizar.

IV.3.2.1. Planificación

En esta iteración se realizó el levantamiento de información respecto a la base de conocimientos que servirán de entrada a la red neuronal a implementar, como estrategia de detección de ojos afectados por retinoblastoma.

La base de conocimientos que servirán de entrada a la red neuronal, se obtuvo mediante fotografías recreacionales y médicas de los pacientes de la Unidad de Oncología Ocular (UOO) ubicada en el Instituto Oncológico "Dr. Luis Razetti", otra fuente fueron imágenes recreacionales compartidas en la *web Flickr*.

El contenido principal de las imágenes es de pacientes con el síntoma de leucocoria y también se obtuvieron imágenes de pacientes con ojo sano;

eso con el fin de tener una base de conocimientos con casos positivos y casos negativos.

IV.3.2.2. Diseño

A partir de la información y los requerimientos recopilados se procedió a realizar el entrenamiento de la red neuronal artificial. La red neuronal utilizada está basada en la arquitectura *Inception* v3 de *Tensorflow*, la cual se basa en re-entrenar solo la última capa de la red neuronal, permitiendo que el aprendizaje sea más rápido y sencillo.

IV.3.2.3. Implementación

Para el entrenamiento de la red neuronal se tomaron las imágenes recreacionales tomadas a los pacientes de la UOO. Cada imagen capturada procede a ser modificada y separada en dos, ya que, se extrae la región de los ojos; si es positivo para leucocoria se almacenará dicha región recortada de la imagen en una carpeta llamada Retino y si es negativo para leucocoria la región recortada de la imagen será almacenada en una carpeta llamada Normal. Esto se utilizará para clasificar las imágenes en la red neuronal.

Al tener las imágenes clasificadas por tipo dentro de cada carpeta, como se explicó anteriormente, se procede a re-entrenar la última capa de la red neuronal *Inception* v3.

Lo primero es descargar desde el repositorio el código fuente de *Tensorflow*, luego se construye el entrenador para las imágenes; para ello se ejecuta un script el cual carga el modelo de *Inception* v3 pre-entrenado,

luego se procede a eliminar la capa superior antigua y se entrena una nueva con las imágenes clasificadas de los ojos.

La primera fase del proceso mencionado anteriormente, analiza las imágenes deseadas y calcula los valores de cuello de botella para cada una de ellas. El "cuello de botella" es un término informal utilizado para la capa antes de la capa de salida final que realmente es la que hace la clasificación. Esta penúltima capa fue entrenada para dar salida a un conjunto de valores lo suficientemente buenos, para que el clasificador los utilice para distinguir entre las clases que se desea reconozca. Eso significa que se obtendrá un resumen compacto y significante de las imágenes, ya que debe contener suficiente información para que el clasificador realice una buena elección en un conjunto pequeño de valores.

La segunda fase comienza con el entrenamiento real de la capa superior de la red, esto luego de que se completa el procedimiento de los cuellos de botella para cada imagen. En esta fase se observa una serie de salidas, donde se muestran la precisión del entrenamiento, la precisión de la validación y la entropía cruzada. Donde la precisión del entrenamiento se refiere al porcentaje de las imágenes utilizadas en el lote de entrenamiento que fueron etiquetadas correctamente. La precisión de validación, es la precisión en un grupo de imágenes seleccionado aleatoriamente de un conjunto diferente. Y la entropía cruzada, se refiere a una función de pérdida que da una idea de cómo va avanzando el proceso de aprendizaje. El objetivo del entrenamiento es hacer que la pérdida sea lo más pequeña posible.

Al finalizar el proceso de re-entrenamiento de la red, el script generará una nueva versión de la red *Inception* v3 cuya última capa estará re-entrenada con las categorías requeridas por el proyecto y también un archivo con las etiquetas de la clasificación, en este caso Retino y Normal.

IV.3.2.4. Pruebas

Luego de culminar el entrenamiento la red es necesario probar que en efecto la misma funciona correctamente. Para ello, se ejecuta un script indicando los datos de la capa a utilizar, las etiquetas, la imagen a reconocer y la red neuronal. El resultado arrojado será el nombre de la clasificación en la cual está incluida la imagen, sea normal o retino, esto se puede visualizar en la *Figura 9.*

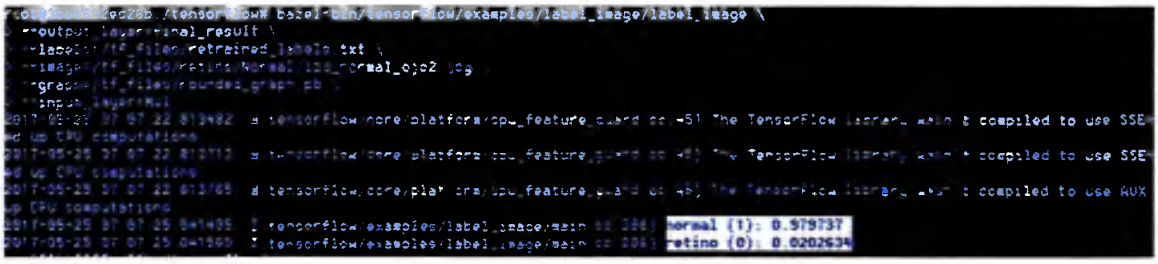

Figura 9. Resultado que arroja la red neuronal cuando se valida una imagen incluida dentro de la clasificación 'normal'

Fuente: Elaboración propia.

IV.3.3. Iteración 3. Módulo de implementación de la red neuronal entrenada.

IV.3.3.1. Planificación

Al inicio de esta iteración se realizó el levantamiento de información correspondiente para el desarrollo e implementación de la red neuronal entrenada en la iteración anterior; de forma que ésta permita el reconocimiento del síntoma de retinoblastoma llamado leucocoria en el ojo de un infante, y arroje un resultado lo más acertado posible.

IV.3.3.2. Diseño

La red neuronal como se mencionó en la iteración anterior, fue realizada utilizando *Tensorflow* para re-entrenar al modelo *Inception* v3; lo cual permite tener una red neuronal entrenada con las clasificaciones que requiere el proyecto. Para la implementación de dicha red neuronal es necesario desarrollar una serie de procedimientos en Java indicados en la implementación que permitan su correcta integración dentro del sistema operativo *Android*.

IV.3.3.3. Implementación

Una vez entrenada la red neuronal con las imágenes y clasificaciones requeridas por el proyecto, es necesario referenciarla dentro de *Android Studio* para que ésta pueda ser utilizada por el dispositivo móvil.

Lo primero es tomar los archivos generados al entrenar la red, como lo son las etiquetas de las clasificaciones y la red misma; ambos se deben

agregar a la carpeta llamada *assets* dentro del proyecto de *Android Studio*. Otro archivo necesario es la librería que permite controlar la API de *Tensorflow* para Java, se debe agregar dentro del proyecto de *Android Studio* en una carpeta llamada *Libs*. Y por último es necesaria la arquitectura de Tensorflow, la cual se guarda en una carpeta de tipo jniLibs con la ruta jniLibs/armeabi-v7a/.

Al tener los archivos mencionados anteriormente en el lugar adecuado, se podrá comenzar a utilizar la API de *Tensorflow* para Java. Se desarrollan las clases para manejar la red neuronal y mediante el uso de la API de *Tensorflow* permita realizar el reconocimiento de la imagen capturada respecto a las almacenadas en la red neuronal.

Dicho reconocimiento comienza con la imagen capturada del ojo, ésta es procesada para ser convertida en datos numéricos y así poder ser introducida en *Tensorflow*; luego se busca dentro de la red neuronal la clasificación de la imagen dentro de las clases definidas y realiza el análisis respecto a las imágenes dada y de la red; y finalmente devuelve el resultado indicando la clasificación dentro de la cual está la imagen capturada por la cámara y el porcentaje de exactitud de dicho resultado.

IV.3.3.4. Pruebas

Luego de culminar con la implementación de la red neuronal dentro de *Android Studio* para que al capturar una imagen con la cámara del dispositivo móvil ésta sea reconocida como normal o con retinoblastoma; es necesario

probar que funciona correctamente dicha implementación; por lo tanto al capturar los ojo de un infante el resultado debe ser dependiendo del estado del ojo, si tiene síntoma de leucocoria, es decir, está afectado por retinoblastoma o si el reflejo es normal. Los resultados de ésta prueba se pueden apreciar en las *Figuras 9, 10 y 11.*

Figura 10. Imagen Original a la cual se le hizo la detección

Fuente: Aplicación desarrollada, LeukoDetect.

Figura 11. Resultado arrojado en el análisis para el ojo izquierdo

Fuente: Aplicación desarrollada, LeukoDetect.

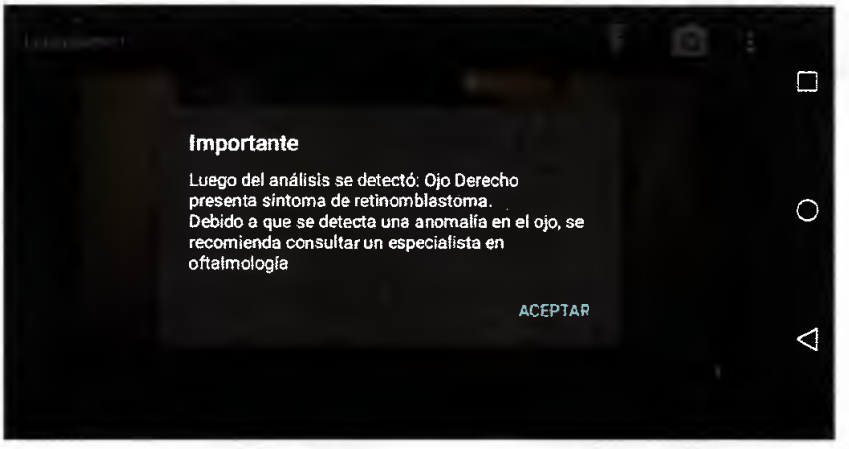

Figura 12. Resultado arrojado en el análisis para el ojo derecho

Fuente: Aplicación desarrollada, LeukoDetect.

Como se puede observar luego de la red neuronal hacer su trabajo, el resultado arrojado es acertado; puesto que en la imagen de la *Figura 9*, se aprecia que en efecto el ojo derecho está afectado por retinoblastoma ya que, la pupila es de coloración blanca y el ojo izquierdo es un ojo normal debido a que el reflejo de la pupila es de color rojizo.

IV.3.4. Iteración 4. Módulo de desarrollo del modelo de datos.

IV.3.4.1. Planificación

Al comienzo de esta iteración se realizó el levantamiento de información requerido para la ejecución de un modelo de datos que permita registrar los resultados arrojados por el sistema, permitiendo obtener estadísticas dentro de la aplicación.

IV.3.4.2. Diseño

Luego de analizar los requerimientos de los datos suministrados, se diseñaron cuatro (4) tablas en la base de datos, en donde se almacenarán los datos necesarios para llevar las estadísticas del análisis de la aplicación. Dichas tablas se puede observar en el siguiente modelo.

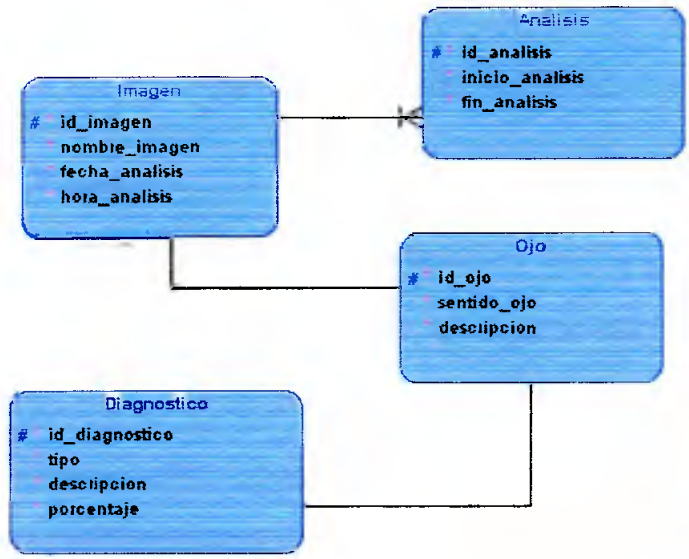

Figura 13. Modelo relacional para el control de las estadísticas de la aplicación

Fuente: Elaboración propia.

IV.3.4.3.Implementacion

Para el desarrollo del modelo de datos, se escogió el motor de base de datos SQLite, el cual permitirá la integración de la base de datos dentro de la aplicación.

Dentro de las tablas del modelo de datos, se almacenará la información de cuándo fue realizado el análisis, el resultado del mismo e información de

la imagen analizada como nombre y ubicación del ojo (derecho o izquierdo); dicha imagen se almacenará en el dispositivo móvil en la memoria externa, dentro de la carpeta de la aplicación y se podrá ubicar en la carpeta de imágenes del *Smartphone*.

Para llevar el control de estadísticas se agregan y consultan los datos en las tablas de la base de datos, permitiendo así reflejar en la aplicación, estadísticas de los últimos 6 (seis) análisis efectuados.

IV.3.4.4. Pruebas

Al finalizar la iteración fue necesario validar que la base de datos está funcionando correctamente, por ende se efectuaron varios análisis en la aplicación y al ingresar en el apartado de control de estadísticas, se verificó que la base de datos está agregando y consultando correctamente. El resultado de dicha prueba se observa en la *Figura 14.*

Figura 14. Control de estadísticas de los análisis efectuados por la aplicación

Fuente: Aplicación desarrollada, LeukoDetect.

CAPÍTULO V

RESULTADOS

En este capítulo se presentan los resultados obtenidos a partir de la aplicación de la metodología XP en el desarrollo de este Trabajo Especial de Grado.

Al finalizar el ciclo de vida del proyecto, a través de las iteraciones dictadas por la metodología XP, con el fin de verificar el cumplimiento de los objetivos planteados, se analizarán uno a uno y se presentará a continuación los resultados logrados en cada uno de ellos.

- **Definir y desarrollar el método de detección de rostros de infantes desde la cámara del teléfono en tiempo real.**

Para lograr este objetivo, luego de investigar las opciones disponibles, se optó por utilizar OpenCV, esta librería *open source*, cuenta con una serie de algoritmos los cuales ayudan a que la detección se realice de forma sencilla. Al utilizar el clasificador entrenado para la detección de rostros de OpenCV, se obtuvo como resultado al utilizar la cámara del dispositivo móvil la detección del rostro en tiempo real y de forma acertada.

- **Definir y desarrollar el método de detección de los ojos en el rostro del infante en tiempo real desde la cámara del teléfono.**

Este objetivo y el anterior están estrechamente relacionados, puesto que la detección ojos solo será posible si se detecta un rostro.

Para lograr detectar los ojos también se hizo uso de la librería OpenCV, utilizando los algoritmos con los que cuenta, los cuales facilitan la detección. Al utilizar el clasificador entrenado para la función de detección de ojos, se obtuvo como resultado al utilizar la cámara los ojos detectados en tiempo real y de forma acertada.

- **Definir y desarrollar el método de detección de iris y pupilas en los ojos detectados del infante en tiempo real desde la cámara del teléfono.**

A partir de la detección del rostro y los ojos, es necesario a ubicar en los ojos el iris y las pupilas basándose en las características típicas como lo el patrón circular y pigmentación oscura o clara en el caso de la pupila. Como resultado final se obtuvo la detección de las pupilas e iris en los ojos capturados con la cámara del dispositivo móvil.

- **Definir las características a codificar que constituyen la base de conocimientos de entrada de la red neuronal artificial a implementar, como estrategia de detección y reconocimiento de retinas afectadas por retinoblastoma.**

Para lograr este objetivo, se tomó en cuenta un síntoma muy común en el retinoblastoma, que es la presencia de leucocoria en el ojo. Por esta razón las características se dividieron en imágenes de ojos con leucocoria, donde la pupila es color blanca, y en imágenes de ojos normales donde la pupila es de color negro o el reflejo que se observa es de color rojo el cual es normal.

- **Desarrollar la red neuronal artificial mencionada en el punto anterior, capaz de establecer el patrón que identifique en imagen las características del Retinoblastoma.**

Para el desarrollo de la red neuronal artificial, fue necesario utilizar una red que permitiera el reconocimiento de imágenes; por lo tanto se utilizó *Inception v3* de *Tensorflow*; esta red neuronal permite el re-entrenamiento de su última capa con las imágenes y clasificaciones requeridas por este proyecto. El patrón establecido para identificar las características del retinoblastoma es que en la imagen se encuentren en la pupila los *píxeles* más claros, indicando la presencia de leucocoria. En caso de ser positiva la

presencia de leucocoria la imagen es clasificada como Retino, en caso contrario como Normal.

- **Diseñar y desarrollar el modelo de datos que de soporte a la persistencia de información.**

En base a los requerimientos del sistema, se diseñó una estructura de datos que permite almacenar los resultados arrojados luego del análisis, permitiendo al usuario consultar las estadísticas de los últimos seis (6) análisis realizados por la aplicación.

- **Diseñar y desarrollar el modelo de interacción hombre-máquina para que la aplicación sea intuitiva y de fácil uso.**

Para lograr este objetivo, se diseñó una interfaz de fácil uso, donde al iniciar la aplicación, se muestra inmediatamente la imagen que obtiene cámara del dispositivo e inicia con el *flash* encendido, ya que permite que mayor visibilidad de la leucocoria. Cuenta en la barra superior de menú, con un botón para apagar el *flash* o encender el *flash* y otro botón para capturar la imagen de los ojos a ser analizada. También cuenta con un botón de configuración donde se muestra más información de la aplicación, las estadísticas de los análisis realizados y una breve explicación de cómo utilizarla, versión y autor.

CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

VI.1. CONCLUSIONES

A partir del cumplimiento de los objetivos establecidos en este Trabajo Especial de Grado y una vez realizado el análisis de los resultados obtenidos, se pueden plantear las siguientes conclusiones:

- La aplicación del algoritmo de redes neuronales de *data mining* como técnica de análisis de datos del proyecto fue la decisión más acertada; ya que permite descubrir patrones en grandes cantidades de datos, logrando así su objetivo de encontrar comportamientos predictivos en éstos.
- Para que las predicciones sean precisas se necesita un modelo de conocimiento con suficiente data, se podría hablar de un gran volumen de datos; para lograr esto se debe ajustar el almacenamiento de esos datos, organizarlos para ser indexados, y que el acceso sea lo suficientemente rápido permitiéndoles escalar. En ese contexto, se puede hacer una analogía entre los datos del modelo de conocimiento y las bases de datos NoSQL.
- Se puede concluir que utilizar redes neuronales convolucionales en vez de las redes neuronales convencionales fue lo adecuado, debido a que éste tipo de redes suponen explícitamente imágenes como datos de entrada, lo cual se adapta a las necesidades del sistema.

Adicionalmente son capaces de efectuar modelos complejos dando como resultado predicciones más precisas.

- El uso de la librería de Google *Tensorflow* con la arquitectura de *Inception* v3 permitió que el entrenamiento y desarrollo de la red neuronal fuese comprensible, sencillo y eficaz. Puesto que *Inception* v3 brinda la facilidad de solo entrenar su última capa, y las capas inferiores que han sido entrenadas para distinguir entre algunos objetos pueden ser reutilizadas para muchas tareas de reconocimiento sin ninguna alteración.

VI.2. RECOMENDACIONES

Aunque la solución engloba los objetivos y alcance planteados, a continuación se listan algunos aspectos adicionales que pueden ser considerados en investigaciones futuras con el fin de obtener mejores resultados:

- Implementar la adición de imágenes a la base de conocimientos que conforman la red neuronal artificial directamente desde la aplicación, es decir, que el usuario al capturar la imagen de los ojos de un paciente tenga la opción de registrarla para que esta sea utilizada en la retroalimentación de versiones posteriores del sistema.
- El proyecto solo puede ser utilizado actualmente en dispositivos móviles cuyo sistema operativo sea *Android*; se recomienda la implementación de una plataforma de *web service* o centralizada, con el fin de que todo el proceso recaiga sobre un servidor y allí sea analizado, abriendo la posibilidad de escalabilidad de la aplicación en diversos sistemas operativos móviles.
- Se recomienda enriquecer la data mediante el uso de estrategias como entrevistas a los representantes de pacientes con retinoblastoma; generando un proceso de retroalimentación, el cual permitirá que la red se reconfigure y obtenga el modelo adecuado con un porcentaje de exactitud de 100%.

GLOSARIO

Cáncer: El cáncer es un proceso de crecimiento y expansión incontrolado de células. Puede aparecer en cualquier parte del cuerpo. El tumor por lo general invade el tejido adyacente y puede provocar metástasis en distintas partes del cuerpo. Muchos tipos de cáncer se podrían prevenir evitando exponerse a factores de riesgo como el humo de tabaco. Además, un gran porcentaje de cánceres se pueden curar mediante cirugía, radioterapia o quimioterapia, y más si se detectan en una fase temprana.

Infante: Infante, del latín *infans* o *infantis* "el que no habla", según el diccionario de la Real Academia Española, es un niño de corta edad. Infante, es el término legal para referirse a los menores de edad, que incluye a los niños de 0 (cero) a 5 (cinco) años de edad.

Leucemia: La leucemia es una enfermedad cancerígena del sistema productor de sangre. Es poco frecuente en comparación con otros tipos de cáncer. La leucemia se interpreta como un aumento desmesurado de los glóbulos blancos (leucocitos) en la médula ósea. Sin embargo, no son maduros ni funcionales. Esta situación altera la formación normal de sangre y lleva a una falta de leucocitos, glóbulos rojos (eritrocitos) y plaquetas (trombocitos) maduros y sanos. Existen cuatro tipos de

leucemia según su evolución, se dividen en leucemias agudas y crónicas:

- La leucemia aguda se caracteriza por el rápido desarrollo de la enfermedad.
- La leucemia crónica tiene un desarrollo lento y silencioso. El paciente tarda meses o años en presentar síntomas graves.

Otra clasificación se basa en el grado de madurez (células maduras e inmaduras):

- Leucemia linfocítica: las células de leucemia surgen por la degeneración de otra familia de células, los linfocitos.
- Leucemia mieloide: los leucocitos degenerados descienden del tejido de la médula ósea, del que se extienden varias células sanguíneas, incluyendo los granulocitos.

Leucocoria: La leucocoria procede del griego (leuko: blanco y coria: pupila) lo cual se refiere a tener una pupila de aspecto blanco en lugar del habitual color negro. Las causas más frecuentes de leucocoria son catarata pediátrica, retinoblastoma, traumatismos, infecciones e inflamaciones.

Meduloblastoma: El meduloblastoma es un cáncer que se desarrolla en el cerebelo, la parte posterior e inferior del cerebro, la cual es responsable del movimiento, balance, coordinación y algunas funciones cognitivas; afecta mayormente a niños. Las causas exactas

no se conocen, pero como el Meduloblastoma ocurre principalmente en la niñez, se sospecha que hay un papel clave en los agentes que operan muy temprano en la vida o durante el embarazo materno, como los productos químicos tóxicos, algunos alimentos o las infecciones.

Nefroblastoma: El nefroblastoma, conocido como tumor de Wilms (WT) por sus siglas en inglés, es un raro cáncer infantil en el riñón, sin embargo es el más común de los cáncer de hígado en niños y adolescentes siendo aproximadamente el 90% (noventa por ciento) de los tumores renales en menores de 15 (quince) años; está asociado a una proliferación anormal de células parecidas a las células del riñón del embrión (metanefroma), de dónde viene el término de tumor embrionario. El nefroblastoma afecta principalmente a niños pequeños, entre 1 (uno) y 5 (cinco) años, pero un 15% (quince por ciento) de los nefroblastomas ocurren antes del primer año de edad y un 2% (dos por ciento) después de los 8 (ocho) años. Este tumor es causado por mutaciones genéticas, muchas de las cuales son desconocidas; algunas pueden ser producidas por exposición a agentes como pesticidas antes del nacimiento (durante el embarazo) o justo después, o incluso por exposición de los padres antes de concebir.

Neuroblastomas: El neuroblastoma es una enfermedad por la que se forman células malignas a partir de los neuroblastos (tejido nervioso inmaduro) de la glándula suprarrenal, el cuello, el tórax o la médula espinal. El neuroblastoma generalmente comienza en el primer año de

vida y es posible diagnosticarlo durante el primer mes. En algunos casos, se forma antes del nacimiento y se encuentra durante una ecografía del feto. Comúnmente la causa del neuroblastoma es una mutación génica (cambio) que se trasmite de los padres a los hijos.

Proteína Rb: La proteína Rb (también denominada pRB) es la proteína del retinoblastoma, actúa al alterar la actividad de los factores de transcripción. Al interactuar con los factores de transcripción, Rb es capaz de controlar la expresión genética indirectamente. La proteína Rb y sus parientes contribuyen al control del proceso de la división celular. El gen Rb se encuentra mutado en varios tipos de cánceres. El más investigado es el retinoblastoma, del cual el gen obtuvo su nombre.

SQL: Son las siglas en inglés de Structured Query Language (Lenguaje de Consulta Estructurado), es el lenguaje que da acceso al manejo de bases de datos de carácter relacional y permite la realización de distintas operaciones entre éstas. Gracias al uso del álgebra y de cálculos relacionales, SQL permite efectuar consultas con el fin de recuperar información de las bases de datos de manera sencilla, así como también hacer cambios en ellas.

BIBLIOGRAFÍA

¿Qué es el cáncer y cómo se desarrolla? (2016). Seom.org. Consultado el 11 Jun 2016, de http://www.seom.org/en/informacion-sobre-el-cancer/que-es-el-cancer-y-como-se-desarrolla

¿Qué es el cáncer? (2016). National Cancer Institute. Consultado el 11 Jun 2016, de http://www.cancer.gov/espanol/cancer/que-es

¿Qué Es un Retinoblastoma? (2012). American Academy of Ophthalmology. Consultado el 5 May 2017, de https://www.aao.org/salud-ocular/enfermedades/que-es-retinoblastoma.

"Para prevenir la Retinoblastoma, se debe realizar un examen a los recién nacidos". (2010). Mppeuct.gob.ve. Consultado el 2 May 2017, de http://www.mppeuct.gob.ve/actualidad/noticias/para-prevenir-la-retinoblastoma-se-debe-realizar-un-examen-los-recien-nacidos

A, C. (2013). Detección de rostros. Acodigo.blogspot.com. Consultado el 7 May 2017, de http://acodigo.blogspot.com/2013/06/deteccion-de-rostros.html

About SQLite. Sqlite.org. Consultado el 24 May 2017, de http://www.sqlite.org/about.html

ACS :: What Are the Key Statistics for Retinoblastoma?. (2008). Web.archive.org. Consultado el 4 Jul 2016, de http://web.archive.org/web/20090531021149/http://www.cancer.org/docroot/CRI/content/CRI_2_4_1X_What_are_the_key_statistics_for_retinoblastoma_37.asp?sitearea=

Bautista Q., J. (2016). PROGRAMACIÓN EXTREMA (XP) EXTREME PROGRAMMING (XP) (1st ed.). Universidad Unión Bolivariana. Consultado el 3 Jul 2016, de http://ingenieriadesoftware.mex.tl/images/18149/PROGRAMACI%C3%93N%20EXTREMA.pdf

Baviera, C., tratamientos, O., & divergente, E. (2016). Estrabismo convergente y divergente: diferencias | Clínica Baviera. Clinicabaviera.com. Consultado el 1 Jul 2016, de http://www.clinicabaviera.com/estrabismo-divergente-y-convergente

Beck, K., & Fowler, M. (1999). Planning extreme programming (1st ed.). Boston: Addison-Wesley.

Briega, R. (2016). TensorFlow y Redes Neuronales. Relopezbriega.github.io. Consultado el 5 May 2017, de http://relopezbriega.github.io/blog/2016/06/05/tensorflow-y-redes-neuronales/

Briega, R. (2017). Redes neuronales convolucionales con TensorFlow. Relopezbriega.github.io. Consultado el 5 May 2017, de http://relopezbriega.github.io/blog/2016/08/02/redes-neuronales-convolucionales-con-tensorflow/

Cáncer. (2017). Organización Mundial de la Salud. Consultado el 5 May 2017, de http://www.who.int/topics/cancer/es/

Cascade Classification — OpenCV 2.4.13.2 documentation. (2017). Docs.opencv.org. Consultado el 11 May 2017, de http://docs.opencv.org/2.4/modules/objdetect/doc/cascade_classification.html

Childhood Medulloblastoma - 2015 - Startoncology. (2015). Startoncology.net. Consultado el 19 May 2017, de http://www.startoncology.net/public-area/childhood-medulloblastoma-2015/?lang=en

Estadísticas del cáncer. (2016). National Cancer Institute. Consultado el 4 Jul 2016, de http://www.cancer.gov/espanol/cancer/naturaleza/estadisticas

Extreme Programming (XP). (2016). Consultado el 3 Jul 2016, de www.fing.edu.uy/inco/cursos/gestsoft/Presentaciones/XP%20-%20G3/XP.ppt

Gracia, L. (2013). ¿Qué es OpenCV?. Un poco de Java. Consultado el 5 May 2017, de https://unpocodejava.wordpress.com/2013/10/09/que-es-opencv/

Image classification with a pre-trained deep neural network. (2016). Kernix.com. Consultado el 6 May 2017, de https://www.kernix.com/blog/image-classification-with-a-pre-trained-deep-neural-network_p11

Julián, G. (2014). Las redes neuronales: qué son y por qué están volviendo. Xataka.com. Consultado el 1 Jul 2016, de http://www.xataka.com/robotica-e-ia/las-redes-neuronales-que-son-y-por-que-estan-volviendo

Leucemia (cáncer de la sangre) - Onmeda.es. (2016). onmeda.es. Consultado el 19 May 2017, de http://www.onmeda.es/enfermedades/leucemia.html

LÓPEZ MECA, I., & RUIZ BELDA, C. (2014). Leucocoria o pupila blanca en el niño - SOCV - Sociedad Oftalmológica de la Comunidad Valenciana. SOCV - Sociedad Oftalmológica de la Comunidad Valenciana. Consultado el 19 May 2017, de http://www.socv.org/leucocoria-o-pupila-blanca-en-el-nino/

Moreno, A. (2017). Redes Neuronales Artificiales Arquitecturas y Aprendizaje. Academia.edu. Consultado el 5 May 2017, de

http://www.academia.edu/8172191/Redes_Neuronales_Artificiales_Arquitecturas_y_Aprendizaje

Neuroblastoma. (2017). National Cancer Institute. Consultado el 19 May 2017, de https://www.cancer.gov/espanol/tipos/neuroblastoma/paciente/tratamiento-neuroblastoma-pdq

Pérez Porto, J., & Gardey, A. (2013). Definición de Java — Definicion.de. Definición.de. Consultado el 5 May 2017, de http://definicion.de/java/

Pérez Porto, J., & Gardey, A. (2012). Definición de SQL — Definicion.de. Definición.de. Consultado el 24 May 2017, de http://definicion.de/sql/

Qué es un kit de desarrollo de software (SDK)? | 4R Soluciones | Diseño, Desarrollo y Programación Web & Mobile. (2013). 4R Soluciones | Diseño, Desarrollo y Programación Web & Mobile. Consultado el 3 Jul 2016, de http://www.4rsoluciones.com/blog/que-es-un-kit-de-desarrollo-de-software-sdk-2/

Real Academia Española. Diccionario Usual. (2017). Infante,ta. Consultado el 5 May 2017, de http://dle.rae.es/srv/fetch?id=LUdK633.

RETINOBLASTOMA - Union for International Cancer Control - Review of Cancer Medicines on the WHO List of Essential Medicines. (2014) (1st ed.).

Retinoblastoma. (2016). National Cancer Institute. Consultado el 2 May 2017, de https://www.cancer.gov/espanol/tipos/retinoblastoma/paciente/tratamiento-retinoblastoma-pdq

Roca, D. (2016). RETINOBLASTOMA: ¡OJO CON ESTE TUMOR MALIGNO! Anmdecolombia.net. Consultado el 21 Jun 2016, de http://anmdecolombia.net/index.php/50-home/noticias1/sesiones/298-ojo-con-el-retinoblastoma

Rómmel, F. (2007). SQLite: La Base de Datos Embebida | SG. Sg.com.mx. Consultado el 24 May 2017, de https://sg.com.mx/revista/17/sqlite-la-base-datos-embebida#.WSUTFpLhDIV

Ruiz, S. (2015). La predicción del dato: Redes Neuronales Artificiales | Analítica web. Analiticaweb.es. Consultado el 4 Jul 2016, de http://www.analiticaweb.es/la-prediccion-del-dato-redes-neuronales-artificiales/

Sanchez, C. (2004). Ciclo de vida de un proyecto XP. Oness.sourceforge.net. Consultado el 10 May 2017, de http://oness.sourceforge.net/proyecto/html/ch05s02.html

Solis, C. (2014). Android SDK: Guía de instalación para Mac y Windows. Revolución móvil. Consultado el 3 Jul 2016, de http://revolucion.mobi/2014/02/11/guia-instalacion-android-sdk/

Steliarova-Foucher, E., & Ullrich, A. (2017). Preguntas frecuentes sobre el cáncer infantil. Organización Mundial de la Salud. Consultado el 5 May 2017, de http://www.who.int/cancer/media/news/Childhood_cancer_day/es/

Trujillano, J., March, J., & Sorribas, A. (2004). Aproximación metodológica al uso de redes neuronales artificiales para la predicción de resultados en medicina. Consultado el 4 Jul 2016, de http://www.elsevier.es/en-revista-medicina-clinica-2-articulo-aproximacion-metodologica-al-uso-redes-13057536http://www.elsevier.es/en-revista-medicina-clinica-2-articulo-aproximacion-metodologica-al-uso-redes-13057536

Venezuela registra más de 30 millones de suscriptores de telefonía móvil. (2015). CONATEL. Consultado el 4 Jul 2016, de http://www.conatel.gob.ve/venezuela-registra-mas-de-30-millones-de-suscriptores-de-telefonia-movil/

Versión en inglés revisada por: Adam S. Levy, a. (2016). Retinoblastoma: MedlinePlus enciclopedia médica. Nlm.nih.gov. Consultado el 21 Jun 2016, de https://www.nlm.nih.gov/medlineplus/spanish/ency/article/001030.html

Villasana, M. (2017). Introducción a Las redes Neuronales (Neurales) CO-6612. Consultado el 5 May 2017, de http://prof.usb.ve/mvillasa/redes/Introduccion-vl.pdf

Zamora, E. (2016). Redes Neuronales Convolucionales. Scribd. Consultado el 5 May 2017, de https://es.scribd.com/doc/295974900/Redes-Neuronales-Convolucionales

APÉNDICES

Apéndice 1. Historias de usuario

Código	Nombre
HU-1	Detectar rostros
HU-2	Detectar ojos
HU-3	Detectar iris y pupila
HU-4	Entrenar red neuronal
HU-5	Implementar red neuronal
HU-6	Crear estructura de datos

Fuente: Elaboración propia

Apéndice 2. Plan de entregas

Código	Primera Iteración	Segunda Iteración	Tercera Iteración	Cuarta Iteración
HU-1				
HU-2				
HU-3				
HU-4				
HU-5				
HU-6				
	12 Semanas	10 Semanas	10 Semanas	4 Semanas

Fuente: Elaboración propia