

Universidades de Burgos, León y
Valladolid

Máster universitario

Inteligencia de Negocio y Big Data en Entornos Seguros



**TFM del Máster Inteligencia de Negocio
y Big Data en Entornos Seguros**

Clasificación del iris

Presentado por Ignacio Ponsoda Llorens
en Universidad de Burgos — 14 de junio
de 2022

Tutor: Dr. José Francisco Díez Pastor y Dr.
Pedro Latorre Carmona

Universidades de Burgos, León y Valladolid



Máster universitario en Inteligencia de Negocio y Big Data en Entornos Seguros

D. José Francisco Díez Pastor, profesor del departamento de Ingeniería Informática, área de Lenguajes y Sistemas Informáticos. D. Pedro Latorre Carmona, profesor del departamento de Ingeniería Informática, área de Lenguajes y Sistemas Informáticos.

Expone:

Que el alumno Ignacio Ponsoda Llorens, con DNI 21698927Z, ha realizado el Trabajo final de Máster en Inteligencia de Negocio y Big Data en Entornos Seguros titulado título de TFM.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 14 de junio de 2022

Vº. Bº. del Tutor:

Vº. Bº. del tutor:

D. José Francisco Díez Pastor

D. Pedro Latorre Carmona

Resumen

La utilización de la biometría para mejorar la seguridad de los aparatos electrónicos es un recurso ampliamente utilizado hoy en día. El iris es uno de los elementos biométricos con los que cuenta el ser humano. Este trabajo analiza la capacidad de reconocimiento de individuos que tienen las redes neuronales personalizadas. Para ello, utilizando *fine-tuning*, se han adaptado cuatro variaciones del *dataset* a la red VGG16. Dos de estas variaciones corresponden, por un lado, al *dataset* sin preprocesos y por otro, a imágenes donde se ha segmentado y normalizado el iris. Las dos variaciones restantes, han sido determinadas por la utilización de *data augmentation* sobre las dos descritas anteriormente.

Los resultados muestran que la red neuronal ha funcionado de forma más eficiente en las imágenes donde no se había aislado la zona del iris. Así mismo, la red ha tenido una mejor capacidad de clasificación cuando no se le ha aplicado el *data augmentation*, algo que puede deberse a que las imágenes no han sido testadas en situaciones reales.

En base a estos resultados, se puede extraer la capacidad de las propias redes neuronales para encontrar características del ojo humano más allá del iris, y así mejorar su capacidad de clasificación.

Descriptores

biometría, iris, redes neuronales, *fine-tuning*, *data augmentation*

Abstract

The use of biometrics to improve the security of electronic devices is a widely used resource nowadays. The iris is one of the biometric elements available to the human being. This work analyzes the recognition capacity of individuals of adapted neural networks. To do this, using fine-tuning, four variations of the dataset have been adapted to the VGG16 network. Two of these variations correspond, on the one hand, to the dataset without preprocessing, and on the other, to images where the iris has been segmented and normalized. The two remaining variations have been determined by the use of data augmentation on the two previously described.

The results show that the neural network has worked more efficiently in the images where the iris area had not been isolated. Likewise, the network has had a better classification capacity when the data augmentation has not been applied, something that can This may be because the images have not been tested in real scenarios.

Based on these results, it is possible to extract the capacity of the neural networks themselves to find features of the human eye beyond the iris, and thus improve their classification capacity.

Keywords

biometrics, iris, neural networks, fine-tuning, data augmentation

Índice general

Índice general	iii
Índice de figuras	v
Índice de tablas	vi
Memoria	1
1. Introducción	3
1.1. Outline	5
2. Objetivos del proyecto	7
3. Conceptos teóricos	9
3.1. Biometría	9
3.2. Inteligencia Artificial	10
4. Técnicas y herramientas	19
4.1. Hardware	19
4.2. Github	20
4.3. Python	20
4.4. Scrum	21
5. Aspectos relevantes del desarrollo del proyecto	23
5.1. Preparación del dataset	23
5.2. Preprocesamiento de los datos	23
5.3. Adaptación de la red neuronal	25

6. Trabajos relacionados	29
7. Conclusiones y Líneas de trabajo futuras	31
7.1. Conclusiones	31
7.2. Líneas de trabajo futuras	31
 Apéndices	 32
Apéndice A Plan de Proyecto Software	35
A.1. Introducción	35
A.2. Planificación temporal	35
A.3. Estudio de viabilidad	37
 Apéndice B Especificación de Requisitos	 39
B.1. Introducción	39
B.2. Objetivos generales	39
B.3. Catalogo de requisitos	39
B.4. Especificación de requisitos	39
 Apéndice C Especificación de diseño	 41
C.1. Introducción	41
C.2. Diseño de datos	41
C.3. Diseño procedimental	41
C.4. Diseño arquitectónico	41
 Apéndice D Documentación técnica de programación	 43
D.1. Introducción	43
D.2. Estructura de directorios	43
D.3. Manual del programador	43
D.4. Compilación, instalación y ejecución del proyecto	43
D.5. Pruebas del sistema	43
 Apéndice E Documentación de usuario	 45
E.1. Introducción	45
E.2. Requisitos de usuarios	45
E.3. Instalación	45
E.4. Manual del usuario	45
 Bibliografía	 47

Índice de figuras

1.1. Enfoque utilizando las imágenes sin preprocesamiento.	4
1.2. Enfoque utilizando la normalización y segmentación del iris. . .	5
3.3. Eliminación del reflejo de la pupila.	10
3.4. Ejemplo de <i>data augmentation</i> por ruido gaussiano.	12
3.5. Ejemplo de transformacion de identidad.	13
3.6. Ejemplo de transformacion por reflexión.	13
3.7. Ejemplo de transformacion por escalamiento.	14
3.8. Ejemplo de transformacion por traslación.	14
3.9. Ejemplo de transformacion por rotación.	15
3.10. Ejemplo de segmentación del ojo.	16
3.11. Ejemplo de binarización del ojo durante la segmentación.	16
3.12. Ejemplo de normalización del ojo.	17

Índice de tablas

3.1. Transformaciones afines aplicadas en el trabajo.	12
4.2. Características del equipo	19
5.3. Configuración del <i>pipeline</i>	24
5.4. Tasa de acierto de los modelos.	26

Memoria

Introducción

El *Oxford Learners Dictionaries* ¹ define la biometría como la utilización de características humanas para poder identificar a las personas.

Esta cualidad de la biometría se encuentra principalmente para los humanos en el reconocimiento facial y del iris y huellas dactilares, y hoy en día se ha convertido en un elemento fundamental en la seguridad de nuestros dispositivos electrónicos, lo cual supone indirectamente, que de la biometría depende el acceso a nuestra información privada.

Dentro de la biometría, el iris se utiliza como elemento de reconocimiento biométrico de gran eficacia, tanto por su inmutabilidad a lo largo del tiempo como por resultar un valor único y personal, que supone que dos personas no tendrían un iris idéntico [10].

En este estudio, se han comparado técnicas que permitan la identificación de individuos a través de imágenes de su iris. En concreto, la comparación se ha llevado a cabo, utilizando la tasa de acierto de los modelos a la hora de clasificar.

El proyecto se ha llevado a cabo en dos fases. En una primera fase, se ha empleado la red neuronal VGG16 ², entrenada con ImageNet³, a la que, para que los resultados de clasificación de la red neuronal se adapten a nuestro

¹Definición consultada en <https://www.oxfordlearnersdictionaries.com/definition/english/biometric>

²Esta red neuronal cuenta con 16 capas y ha sido entrenada con más de un millón de imágenes.

³ImageNet es un proyecto donde se proporciona una gran base de datos de imágenes para usos no comerciales <https://www.image-net.org/>

dataset de imágenes, se le ha aplicado *fine-tuning*⁴, tal como se muestra en la imagen 1.1.

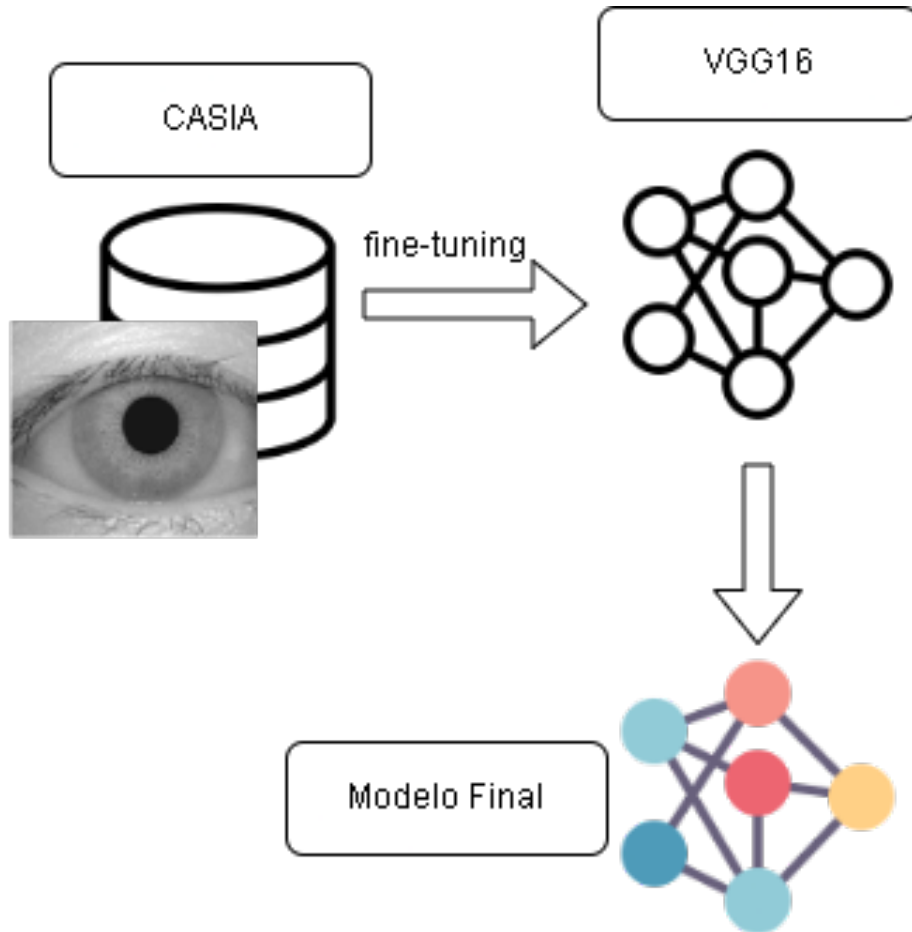


Figura 1.1: Enfoque utilizando las imágenes sin preprocesamiento.

En una segunda fase, se ha aplicado el proceso de extracción y normalización del iris a el mismo *dataset* de imágenes, utilizado en [2], para luego seguir el mismo proceso de adaptación de la red neuronal a el *dataset*, como se muestra en la imagen 1.2.

⁴El *fine-tuning* permite adaptar el modelo para que, al llevarse a cabo la clasificación, no muestre resultados relativos al *dataset* con el que ha sido entrenada, sino con el que ha sido adaptada.

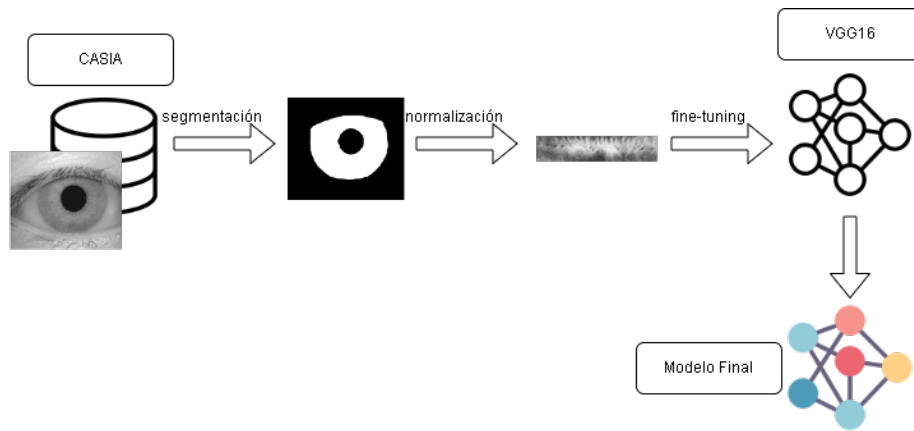


Figura 1.2: Enfoque utilizando la normalización y segmentación del iris.

Así mismo, en ambos enfoques, se han utilizado técnicas de *data augmentation* ⁵ para aumentar el número de imágenes por dataset, algo que a priori mejora la robustez de los modelos.

El objetivo de este proyecto ha sido el de analizar cual de estas perspectivas es más óptima para el reconocimiento de un individuo a través de su iris así como analizar la razón de que la tasa de acierto de una perspectiva sea mayor en uno de los casos.

1.1. Outline

El resto del documento se estructura de la siguiente manera. El capítulo 2 [Objetivos del proyecto](#) define las principales finalidades del proyecto. El capítulo 3 [Conceptos teóricos](#) contiene los conceptos teóricos necesarios para entender el proyecto. El capítulo 4 [Técnicas y herramientas](#) muestra las técnicas y herramientas utilizadas en el desarrollo de este trabajo. En el capítulo 5 [Aspectos relevantes del desarrollo del proyecto](#) se muestran los Aspectos más relevantes que se han desarrollado. En el capítulo 6 [Trabajos relacionados](#) los trabajos relacionados y en el capítulo 7 [Conclusiones y Líneas de trabajo futuras](#) las conclusiones y las líneas de trabajo futuras.

⁵El término *data augmentation* hace referencia a un conjunto de técnicas que permite ampliar el *dataset* original con variaciones de el mismo. En la sección [Data augmentation](#) puede encontrar una explicación más detallada.

Objetivos del proyecto

Son los objetivos en los que principalmente se ha enfocado este trabajo.

- Al tratarse de una continuación de [2], un primer objetivo ha sido el optimizar el código ya existente para mejorar su reproducibilidad y que ello permitiera una mayor flexibilidad a la hora de ejecutar el mismo.
- Utilización de técnicas de *fine-tuning* con el fin de adaptar el dataset a una red neuronal pre-entrenada.
- Aplicación de técnicas de *data augmentation* para comprobar si ello supone una mejora en la robustez del modelo.
- Finalmente, comparar la capacidad de clasificación de los modelos utilizando cuatro variables distintas en su creación. Estas son, la utilización de un dataset donde se utiliza una imagen completa del ojo o una donde el iris ha sido extraído y como afecta la utilización de *data augmentation* en estos modelos.

Conceptos teóricos

En esta sección se explican los principales conceptos teóricos relacionados con el proyecto, relacionados con la biometría, la inteligencia artificial y el *fine-tuning*.

3.1. Biometría

Como se ha comentado en [Introducción](#), la biometría permite la identificación de un individuo a través de determinadas características que se asocian a su persona.

De entre todos los elementos biométricos, que incluyen huellas, cara, iris o voz, el *iris recognition system* (IRS) es el método con mayor eficiencia a la hora de determinar la identidad de las personas [10], ya que el iris es el mismo a lo largo de la vida de una persona y es único, incluso entre gemelos. Esto es incluso utilizado en los procesos post-mortem para poder determinar la pertenencia del cuerpo [5]. El iris humano es un órgano del ojo, que tiene como función, controlar el tamaño de la pupila en función de la cantidad de luz que llega hasta este [5].

Dataset CASIA

Se trata de una base de datos que contiene 756 imágenes del iris de un total de 108 sujetos. Dichas fotos se han realizado en dos sesiones, donde se tomaron 3 y 4 muestras respectivamente por cada individuo. La pupila fue

automáticamente remplazada para evitar que en ella se reflejasen las luces de las fotografías, tal como podemos observar en la figura 3.3 ⁶.

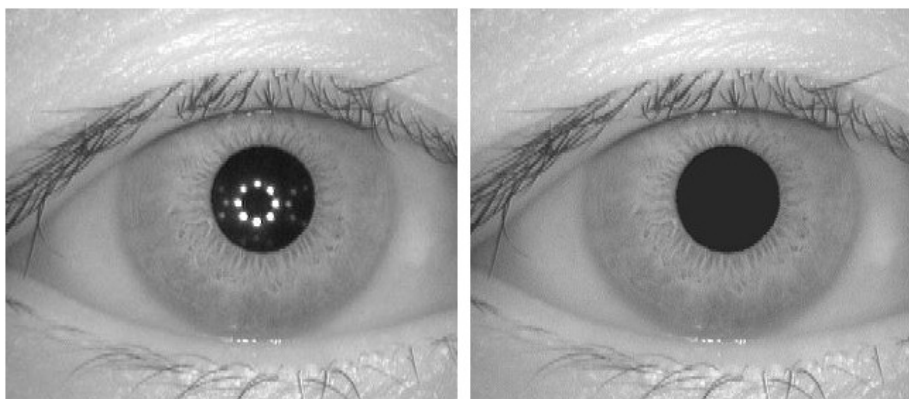


Figura 3.3: Eliminación del reflejo de la pupila [2].

3.2. Inteligencia Artificial

La Real Academia Española define la inteligencia artificial como una disciplina cuyo objetivo principal es la creación de programas capaces de realizar funciones similares a los de la mente humana ⁷.

Deep Learning

El *deep learning* es un tipo de *machine learning*, que utilizando redes neuronales, intenta replicar la forma en la que los humanos toman las decisiones. ⁸

Pipeline

Los *pipelines* son secuencias de procesos encadenados, donde el *output* del proceso anterior funciona como *input* del siguiente proceso. La utilización de *pipelines* ha permitido un mayor control de los distintos procesos, al poder establecer una configuración general para la cadena de procesos, y para poder modificar de una manera más sencilla el orden de los procesos, a la hora de experimentar con nuevas combinaciones.

⁶El proceso de la toma de muestras se describe en <http://www.cbsr.ia.ac.cn/IrisDatabase.htm>

⁷En base a la definición de <https://dle.rae.es/inteligencia>.

⁸En base a la definición de <https://www.ibm.com/cloud/learn/deep-learning>.

Data augmentation

El *data augmentation* es un proceso común en el análisis de imágenes, y en aquellos proyectos donde se utilicen procesos estadísticos.

Este consiste en aumentar el tamaño de los *dataset* con la creación artificial de nuevas imágenes, que son producidas a partir de imágenes del *dataset* original. Para ello, la imagen original se modifica, comúnmente con la aplicación de aumento gaussiano o transformaciones afines, teniendo como resultado, una imagen que deriva de la original, pero que cuenta con ciertas diferencias, que serán más o menos pronunciada dependiendo de las técnicas de *data augmentation* que se le apliquen, así como de los valores utilizados para realizar las modificaciones ⁹.

En el caso de los procesos de entrenamiento de las redes neuronales es común la utilización de técnicas de data augmentation, principalmente por dos situaciones, aunque estas no son limitantes:

- Número insuficiente de datos: en este caso, el data augmentation se aplica porque el dataset no es lo suficientemente grande como para conseguir unos resultados positivos en la creación de una red neuronal.
- Aumento de la robustez del modelo: el segundo supuesto principal por el cual se utiliza data augmentation es la utilización de elementos que añadan dificultades a la red neuronal para cumplir su propósito, lo cual permitirá una mayor robustez del modelo.

Ruido gaussiano

La primera de las técnicas de data augmentation utilizadas ha sido el ruido gaussiano, también conocido como ruido blanco. Esta técnica provoca que los píxeles de una imagen cambien su valor siguiendo una distribución gaussiana, como se puede observar en la figura 3.4 .

⁹Por ejemplo, si se rota una imagen 2 grados, la diferencia con la original será mucho menor que si se rota 180 grados.

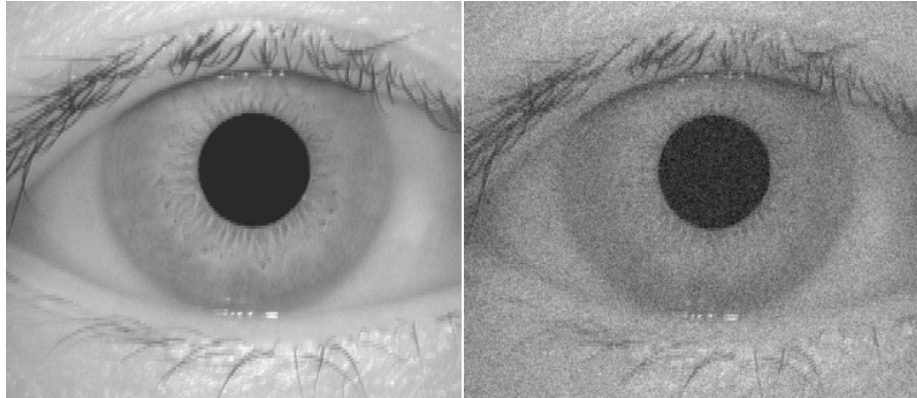


Figura 3.4: Ejemplo de *data augmentation* por ruido gaussiano.

Transformaciones afines

Las transformaciones afines permiten aumentar el tamaño del dataset mediante la transformación de imágenes, donde conservan el paralelismo de sus líneas rectas y paralelas y de alguna forma, simulan una nueva perspectiva de la imagen original. En cuanto a los tipos de transformaciones afines encontramos la transformación de identidad, reflexión, escalamiento, traslación y finalmente, la rotación, como puede comprobarse en la tabla 3.1.

Transformaciones afines				
Identidad	Reflexión	Escalamiento	Traslación	Rotación
Figura 3.5	Figura 3.6	Figura 3.7	Figura 3.8	Figura 3.9

Tabla 3.1: Transformaciones afines aplicadas en el trabajo.

Identidad La transformación de identidad es un tipo de transformación afín en el que la imagen se copia sin ningún otro cambio, y se utiliza para la reutilización de los datasets.

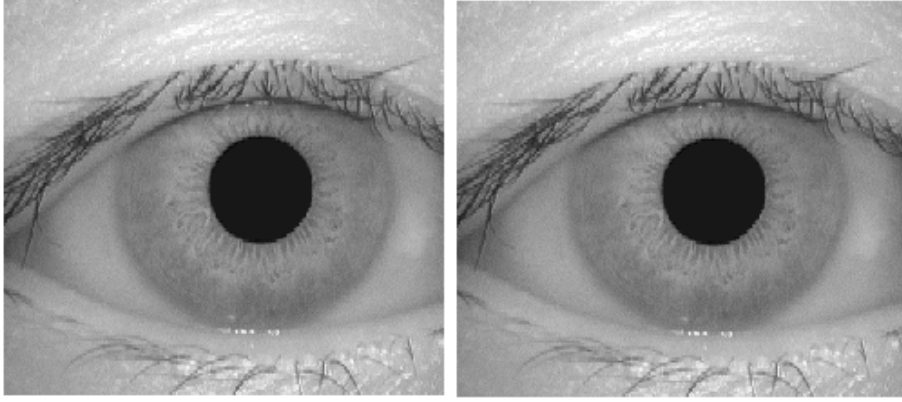


Figura 3.5: Ejemplo de transformacion de identidad.

$$\text{Identidad} : \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \equiv \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (3.1)$$

Reflexión Se trata de un mapeo aplicado a la misma imagen a partir de un eje.

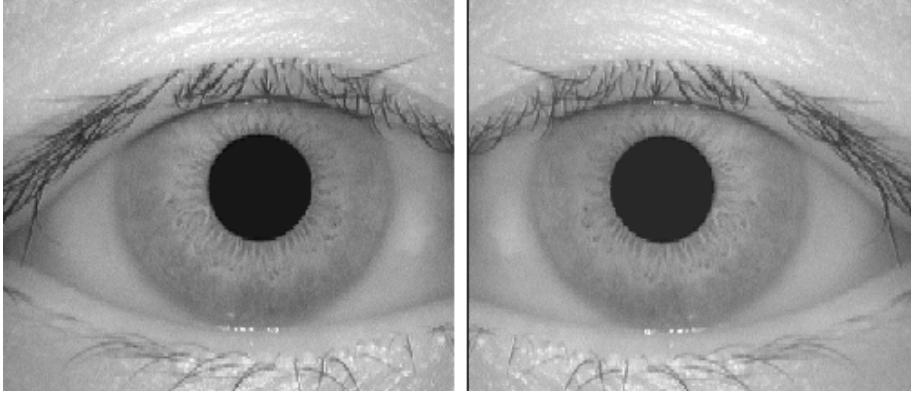


Figura 3.6: Ejemplo de transformacion por reflexión.

$$\text{Reflexión} : \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \equiv \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (3.2)$$

Escalamiento Esta transformación modifica la escala de la imagen original, ya sea ampliándola o disminuyéndola.

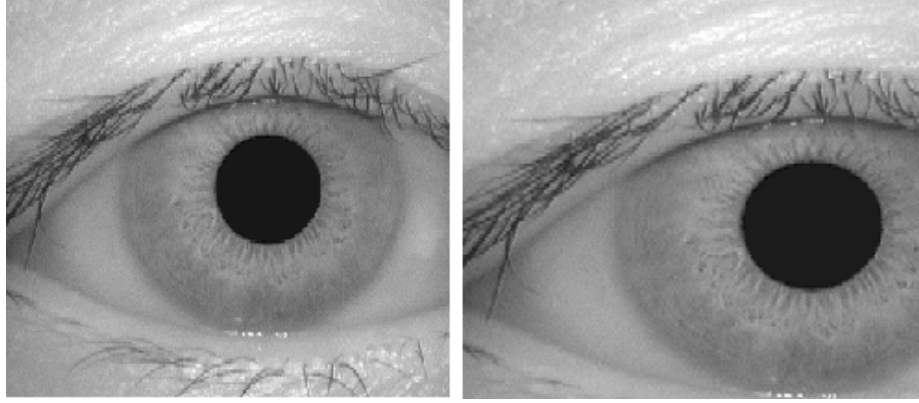


Figura 3.7: Ejemplo de transformación por escalamiento.

$$\text{Escalamiento} : \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \equiv \begin{bmatrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (3.3)$$

Traslación La imagen cambia de plano de coordenadas, pero no se modifican ni su tamaño, ni su forma ni su orientación.

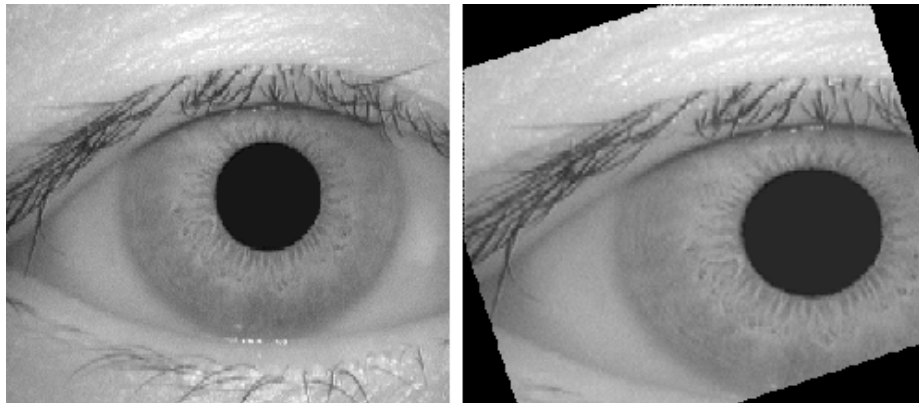


Figura 3.8: Ejemplo de transformación por traslación.

$$\text{Traducción : } \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \equiv \begin{bmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (3.4)$$

Rotación Esta transformación aplica una transformación de θ grados del plano.

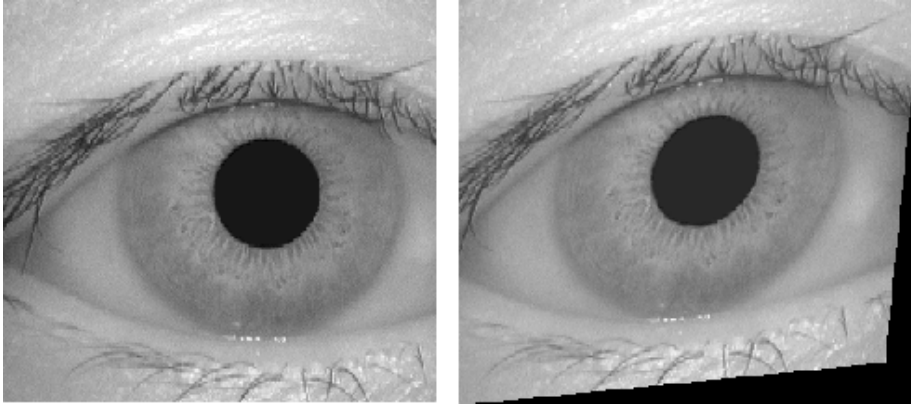


Figura 3.9: Ejemplo de transformacion por rotación.

$$\text{Rotación : } \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \equiv \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (3.5)$$

Preprocesamiento

El preprocesamiento es la manipulación de los datos para que estos tengan el formato requerido para llevar a cabo su procesamiento.

En el caso de este proyecto, la fase de preprocesamiento es la fase en la que se extrae el iris de la imagen, puesto que, tal como indican diferentes estudios [2] [1] [10] [5] [8] [15] [9], la parte del iris es la que permite identificar a las personas.

Segmentación

Detección de los bordes *límico* y *pupilar* utilizando el detector de bordes de Canny [6]. Estos bordes son clave para el aislamiento del iris [2], tal como se observa en la figura 3.10.

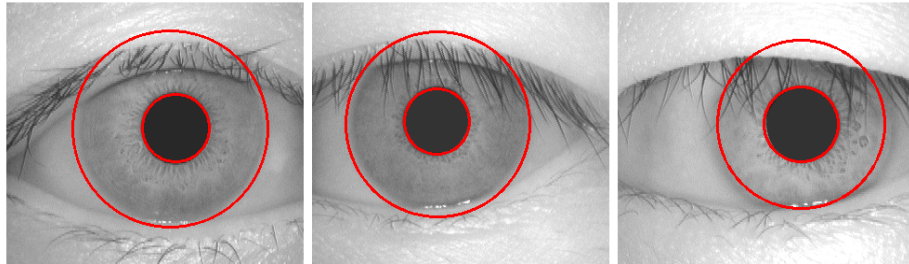


Figura 3.10: Ejemplo de segmentación del ojo extraído de [2].

Una vez detectados los bordes, se procede a una binarización de la imagen 3.11, de forma que quede clara la división entre iris y resto del ojo.

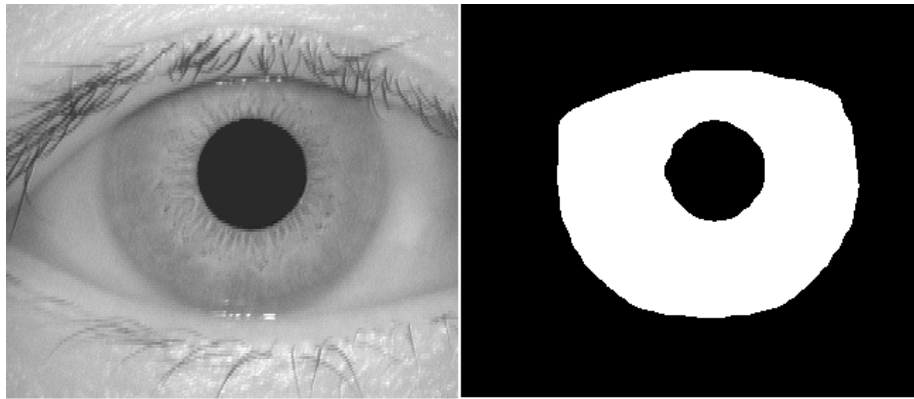


Figura 3.11: Ejemplo de binarización del ojo durante la segmentación.

Normalización

Normalización proyección del iris a coordenadas polares (figura)3.12, utilizando el método Daugman [7], de manera que se igualen los tamaños de las diferentes imágenes y permitan su comparación.

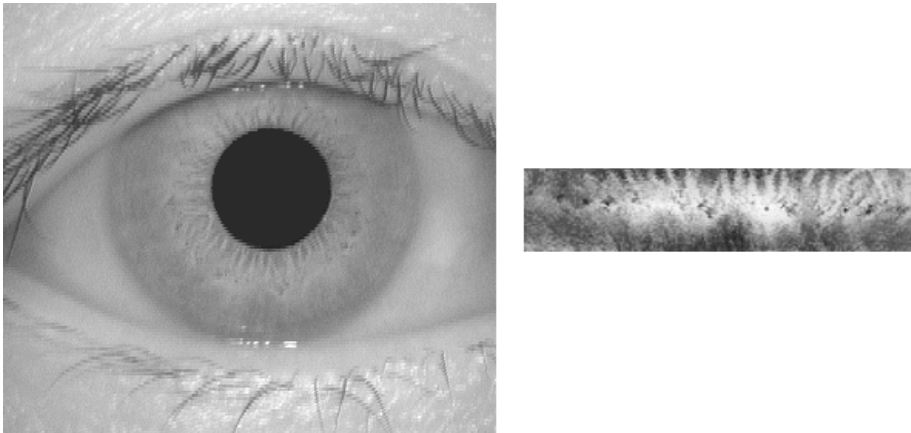


Figura 3.12: Ejemplo de normalización del ojo.

Fine-tuning

Se trata de un método que permite adaptar redes neuronales previamente entrenadas con datos diferentes para un conjunto de datos personalizados. En el caso de este proyecto, el *fine-tuning* ha sido utilizado para obtener un clasificador de imágenes para el dataset de CASIA sin tener que crear una red neuronal desde cero, sino adaptando la red neuronal VGG16.

Técnicas y herramientas

En esta sección se describen los instrumentos y recursos con los cuales se ha llevado a cabo el proyecto.

4.1. Hardware

Para la ejecución de los notebooks de este proyecto se ha contado con una máquina de la UBU, cuyas características se recogen en la tabla 4.2.

Elemento	Característica
Procesador	Intel Xeon E5-2630 v4 @ 2.20GHz (4 núcleos)
Memoria	128GB
GPUs	3 x Titan XP
Discos	SSD 500Gb, 2 x HDD 2TB

Tabla 4.2: Características del equipo

Para interactuar con la máquina, se ha utilizado una conexión SSH ¹⁰ así como un cliente Putty ¹¹, para la modificación de los *notebooks* a través de Jupyter *notebooks* ¹², dentro del ecosistema de Anaconda ¹³.

¹⁰El *security shell* es un protocolo que permite el acceso remoto a través de un canal seguro <https://www.openssh.com/>.

¹¹Putty es una implementación libre de SSH para Windows <https://www.putty.org/>.

¹²Se trata de una aplicación que permite editar y lanzar *notebooks* a través del buscador, <https://jupyter.org/>.

¹³Anaconda es una distribución libre, accesible a través del buscador, que es ampliamente utilizada en la ciencia de datos <https://www.anaconda.com/>.

4.2. Github

Github es una compañía que ofrece repositorios Git¹⁴ en la nube. Estos repositorios se han utilizado en el proyecto¹⁵ para:

1. Control de versiones.
2. Seguimiento de las fases del proyecto.
3. Documentación de las reuniones y de los problemas a resolver.

4.3. Python

Se ha utilizado el lenguaje de programación **Python** para las distintas fases del proyecto.

Entre las principales librerías utilizadas se encuentran:

- **imageio** - leer y escribir imágenes.
- **keras** - manejo de las redes neuronales.
- **matplotlib** - visualización de datos e imágenes.
- **numpy** - trabajo con las imágenes a nivel de arrays.
- **os** - acceso a los directorios.
- **opencv** - trabajo con las imágenes.
- **scikit-image** - transformación de las imágenes y el uso de dataset.
- **shutil** - copia de directorios.
- **tensorflow** - modificación de las redes neuronales.

¹⁴Sistema de control de versiones <https://git-scm.com/>.

¹⁵Repositorio del proyecto es accesible desde <https://github.com/Ponsoda/tfm-iris-recognition>.

Redes neuronales

En el proyecto, dos redes neuronales han sido utilizadas. En primer lugar, en el código optimizado [2], se ha utilizado la red neuronal previamente entrenada, basada en U-Net¹⁶ y accesible desde el [repositorio](#) del *paper* [9]. Esta red neuronal, ya había sido entrenada para la segmentación del iris, por lo que se pudo utilizar directamente para dicha fase, al proveerle de imágenes del *dataset*.

Para el proceso de *fine-tuning*, se ha utilizado la red neuronal VGG16, que ha sido entrenada con el *dataset* de ImageNet.

Visual Studio Code

Tanto para la redacción de la memoria con \LaTeX como para el la creación de los *notebooks* a nivel local, se ha utilizado [Visual Studio Code](#).

Se trata de un IDE¹⁷ con licencia *open-source*, desarrollado por Microsoft que permite funciones de desarrollo, como la edición de código o su depuración.

4.4. Scrum

Scrum es un marco que ayuda a la organización de los equipos entorno a un proyecto, en base a Sprints de una determinada duración y permite una retroalimentación continua del proyecto, de forma que se asegura que el equipo trabaje en consonancia.

En el caso de este proyecto, se ha adecuado la metodología de Scrum para cuadrar reuniones semanales o bisemanales con los tutores del proyecto, estableciendo los sprints y los objetivos de cada sprint utilizando la plataforma Github.

¹⁶Red neuronal para la segmentación de imágenes biomédicas desarrollada por la Universidad de Freiburg <https://arxiv.org/abs/1505.04597>.

¹⁷Entorno de desarrollo integrado.

Aspectos relevantes del desarrollo del proyecto

El proyecto se ha dividido en dos fases principales. La primera fase, ha consistido primeramente en la asimilación, adecuación y optimización del código perteneciente al proyecto de partida[2], así como en la aplicación de las técnicas de *data augmentation*, para posteriormente poder utilizar el proceso de normalización del iris en la segunda fase. Por otro lado, en la segunda fase se ha desarrollado la personalización de la red neuronal VGG16 a las imágenes (tanto normalizadas como no procesadas) del *dataset* utilizado en el proyecto.

5.1. Preparación del dataset

El dataset inicial, se ha dividido en un 70 % para llevar a cabo el proyecto, y un 30 % para comprobar la tasa de acierto del modelo para su clasificación. Este dataset del 70 % será el que se utilizará para la creación de modelos, dividiéndose a su vez en 70-30 para ello.

5.2. Preprocesamiento de los datos

En la primera fase del proyecto, se ha llevado a cabo un procesamiento de datos previo.

Nombre	Configuración
Raíz	Establecimiento del directorio raíz
Tamaño dataset	Tamaño de los <i>datasets</i> de entrenamiento y validación
<i>Data augmentation</i>	Tipo a aplicar
Imágenes y gráficas	Mostrar imágenes y gráficas (configurado individualmente)
Modelo	Nombre y ubicación donde guardar la red neuronal
Epochs	Número de epochs por cada red neuronal entrenada
<i>Batch</i>	Tamaño
<i>Random seed</i>	Tamaño
Peso	Peso utilizado para la red neuronal
<i>Fine-tuning</i>	<i>Dataset</i> utilizado

Tabla 5.3: Configuración del *pipeline*

Adecuación del código previo

En el código previo¹⁸, correspondiente a [2] cuenta con un código donde el iris se segmenta y se normaliza.

Durante esta fase, se han llevado a cabo tres aspectos. La primera ha sido la asimilación del *dataset* y del código previo. La segunda ha sido la adecuación del código previo para que fuese funcional,

y seguidamente, se ha llevado a cabo una asimilación del código.

Creación de la *pipeline*

Para el manejo del Código, se ha encapsulado cada sección en el pipeline, y se ha creado un diccionario de configuración, donde se establecen los criterios que se muestran en la tabla 5.3.

Aplicación del *data augmentation* sobre el dataset

La aplicación del data augmentation se ha hecho de forma alternativa, primero con el ruido gaussiano, donde aleatoriamente se aplica ruido gaussiano de 2.5, 5 y 7.5 de desviación.

Además, con las transformaciones afines, que no son excluyentes del ruido gaussiano, ni entre ellas mismas, se aplican aleatoriamente entre cada una de ellas.

¹⁸Accesible desde https://github.com/jaa0124/iris_classifier.

Como resultado, los datasets con data augmentation aplicado cuentan con un total de x imágenes, y más que el dataset original, con una media de z más imágenes por individuo.

5.3. Adaptación de la red neuronal

En la configuración del pipeline se establece a que dataset se aplica el *fine-tuning*. En este caso se ha aplicado a cuatro datasets. Por un lado, a dos datasets sin preprocesamiento, uno con data augmentation y otro sin. Por otro lado, a dos datasets con el preprocesamiento, uno con data augmentation y otro sin.

Aplicación del *fine-tuning*

Para llevar a cabo la aplicación del *fine-tuning*, se han creado 3 redes neuronales.

Primera fase

En primer lugar, se ha creado una primera red neuronal que permita encontrar las características principales de las imágenes. Para ello, se ha eliminado la capa final, correspondiente a la clasificación de la imagen. A la hora de crear el modelo, se establece como 'training=false' para que funcione como un *inference model*.

A la hora de lanzar este primer modelo, que será el modelo base para la red neuronal final, se establece como 'training=false', para mantenerlo en *inference mode*.

Segunda fase

Para la segunda fase, se ha entrenado una nueva capa, que será la que determine el *output*, para luego hacer un *unfreeze* del modelo creado anteriormente.

Tercera fase

En esta tercera fase, se entrena el modelo completo, manteniendo al modelo base como *inference model* para evitar que se vuelva a entrenar.

Clasificación de imágenes

Las imágenes se han clasificado utilizando los modelos creados en la sección anterior, utilizando como dataset de entrada, el 30 % reservado en la primera fase.

Para las imágenes normalizadas, se les aplica la fase de preprocesamiento, para que puedan ser manejadas por el modelo.

Tasa de acierto

La tasa de acierto representa el número de veces que el modelo a determinado correctamente la clase de la imagen.

En la siguiente tabla 5.4, se puede observar la tasa de acierto que han tenido los modelos, a la hora de relacionar las imágenes con los individuos.

	Sin normalización		Con normalización	
Modelo	Con <i>data augmentation</i>	Sin <i>data augmentation</i>	Con <i>data augmentation</i>	Sin <i>data augmentation</i>
<i>Accuracy</i>	0.94298244	0.93421054	0.719704	0.679926

Tabla 5.4: Tasa de acierto de los modelos.

Estos resultados muestran, en primer lugar, que el *data augmentation* no supone sino un decremento del *accuracy*, tanto en los casos donde se normaliza el iris como en los que no. Esto puede deberse a que, aunque el modelo sí tenga una mayor robustez, al calcularse este parámetro utilizando datos sin ninguna modificación, que tienen un gran parecido a las imágenes originales, el *data augmentation* disminuya ligeramente los resultados. No obstante, la mejora de la robustez previsiblemente permitirá al sistema funcionar de forma más eficiente en un contexto no académico.

Por otro lado, en cuanto a los mejores resultados utilizando imágenes no normalizadas, se debe de tener en cuenta que, por la forma en la que funcionan las redes neuronales, en cuanto a la reducción de imágenes para quedarse con sus características más representativas, tiene sentido que al proporcionar más elementos representativos del individuo, y no solo el iris, la propia red neuronal haya sido capaz de encontrar características en la imagen que son más eficientes para su clasificación, y que, de alguna forma, son ajenos a la zona propiamente del iris.

Además, cabe de tener en cuenta que, tal como se explica en [Dataset CASIA](#), el *dataset* utilizado ha sido sometido a un preprocesamiento previo, en el que se eliminó la pupila para evitar que el brillo emitido por las cámaras para tomar las propias imágenes pudiera afectar a la misma. Por lo tanto, realmente la normalización practicada solo ha afectado a la parte exterior al iris, y esto puede haber reducido su efecto.

Trabajos relacionados

El principal apartado anterior se puede encontrar en el TFG de extracción del iris [2], en el cual se basa este trabajo, puesto se realiza a grandes rasgos todo lo relativo a la primera de las opciones del trabajo.

Sobre temas de extracción del iris encontramos [1], donde se hace un preprocesamiento con extracción del iris utilizando Hough Transform y la normalización con Daugmands rubber. Luego, tras eliminar el ruido, la extracción se realiza con transformaciones de wavelet. Finalmente, se crea una red neuronal utilizando el mean-squared error para calcular los pesos en la red.

En [12] encontramos el desarrollo de técnicas de deep learning para el reconocimiento del iris basado en una *convolutional neural network* residual, utilizando una red preentrenada de ResNet50 y fine-tuning, entrenado con una *cross-entropy loss function* (aunque no utilizan *data augmentation*, ni pre-procesan las imágenes, y además, utilizan otro dataset, el IIT Delhi).

En [13], utilizaron el dataset CASIA - 10000 y la arquitectura VGG-Net, lo cual realiza un PCA para extraer los elementos más característicos de las imágenes. Después utilizan algoritmos de clasificación para clasificar las imágenes, como el SVM (esto es similar al TFG) y consiguen unos porcentajes de reconocimiento muy altos.

En [10] se habla de siete pasos en los que se divide un sistema de reconocimiento del iris:

1. adquisición
2. preprocesamiento

3. segmentación
4. normalización
5. extracción de características
6. selección de features únicos y característicos
7. clasificación.

Este *paper* también describe una falta de trabajos entorno a datasets de baja calidad y realza que los sistemas de reconocimiento del iris (IRS) se vuelven poco efectivos cuando las imágenes tienen rotaciones or reflejos, algo que intentamos de mejorar en nuestro proceso, añadiendo ruido con el *data augmentation*.

Este mismo *paper* también comenta los distintos dataset utilizados para estos estudios de reconocimiento de iris, el tipo de ruido utilizado así como su método, los tipos de segmentación tradicional y actual utilizados (habitualmente con redes neuronales), técnicas de normalización y extracción de características, así como los tipos de accuracy de los métodos de iris recognition.

[11] utiliza también el dataset de casia V! (parece que también [14]) para medir el accuracy y [9] junto con Unet para el iris segmentation.

[3] utiliza filtro gaussiano combinado con Hough detección de líneas pero nadie utiliza solo gaussiano y/o transformaciones afines.

[4] utiliza técnicas de deep learning para clasificar imágenes de ojos de el casia iris 300 dataset, no utiliza data augmentation. Para segmentation utilizan una herramienta llamada OSIRIS y prueban deep learning, *fine-tuning* y raw para ver que clasifica mejor. (mejor adaptar una red neuronal con *fine-tuning* antes que crear la nuestra propia.

Conclusiones y Líneas de trabajo futuras

En este apartado se explican las conclusiones así como se establecen las posibles líneas futuras.

7.1. Conclusiones

Como conclusion de este proyecto, la utilización de técnicas de machine learning, contando con ordenadores de pocos recursos, demuestra que el preprocesamiento es necesario para centrar los procesos en los elementos de las imágenes verdaderamente importantes. Por otro lado, una vez que la imagen está pre-procesada, la utilización de redes neuronales con *fine-tuning* para clasificar las imágenes se ha resuelto como un mayor accuracy que el modelo que aplica machine learning en la última fase pero este último ha demostrado ser más rápido, por lo tanto, la utilización de una u otra técnica variará según los recursos que se tengan y el contexto en el que se vaya a utilizar (inmediatez con que se necesiten los resultados).

7.2. Líneas de trabajo futuras

Las líneas de trabajo futuras se podrían determinar con la utilización de estas técnicas con nuevos datasets, consiguiendo un modelo lo suficientemente robusto que permitiese su utilización en un programa de escritorio, con una primera fase de *fine-tuning* con imágenes del ojo del usuario y una segunda fase donde este se utilizase como método de seguridad para el acceso a ciertos documentos de los aparatos electrónicos.

Apéndice

Apéndice A

Plan de Proyecto Software

A.1. Introducción

A.2. Planificación temporal

En la planificación temporal utilizamos SCRUM que se basa en sprints

Sprint 1

- Creación de un repositorio para el control de las versiones y el seguimiento del trabajo.
- Investigación sobre la materia de estudio (antecedentes, casos de uso, estado del arte)
- Investigación sobre posibles dataset de iris para el estudio.
- Investigación sobre la metodología que mejor se adapte al proyecto.
- Estudio sobre el uso de data augmentation sobre datasets de iris.
- Investigación sobre el uso de pipelines para la mejora de la reproducibilidad del trabajo.
- Investigación sobre los transformes de scikit-learn

Sprint 2

- Investigación sobre pipelines que se ejecuten de forma condicional.
- Investigación sobre posibles redes neuronales para la segmentación del iris.
- Investigación para el uso de data augmentation en el paquete Keras.

Sprint 3

- Clarificación de objetivos del proyecto.
- Pruebas con la utilización de ruido gaussiano con standard deviation aleatorio
- Investigación sobre las transformaciones afines como método para el data augmentation.
- Implementación del pipeline.

Sprint 4

- Pruebas con las transformaciones afines de Keras.

Sprint 5

- Pruebas con el transformador de Keras.

Sprint 6 y 7

- Aplicación de fine tuning con redes neuronales.

Sprint 8

- Reconfigurar el pipeline para que el data augmentation solo se aplique al conjunto de entreamiento.
- Prevención de imagenes no segmentadas correctamente para prevenir que entren en el proceso de normalización.

- Incorporación del modelo de deep learning al pipeline.
- Mejora de la configuración del pipeline.

Sprint 9

- Creación de los modelos de clasificación.

A.3. Estudio de viabilidad

Viabilidad económica

Viabilidad legal

Apéndice B

Especificación de Requisitos

- B.1. Introducción
- B.2. Objetivos generales
- B.3. Catalogo de requisitos
- B.4. Especificación de requisitos

Apéndice C

Especificación de diseño

- C.1. Introducción
- C.2. Diseño de datos
- C.3. Diseño procedimental
- C.4. Diseño arquitectónico

Apéndice D

Documentación técnica de programación

- D.1. Introducción
- D.2. Estructura de directorios
- D.3. Manual del programador
- D.4. Compilación, instalación y ejecución del proyecto
- D.5. Pruebas del sistema

Apéndice E

Documentación de usuario

- E.1. Introducción
- E.2. Requisitos de usuarios
- E.3. Instalación
- E.4. Manual del usuario

Bibliografía

- [1] Dr Hadeel N Abdullah. Iris recognition using wavelet transform and artificial neural networks. page 13.
- [2] Johnson Bolívar Arrobo Acaro. Sistema clasificador de iris.
- [3] Kavitha Amit Bakshi, B.G. Prasad, and Sneha K. An efficient iris code storing and searching technique for iris recognition using non-homogeneous k-d tree. In *2015 International Conference on Emerging Research in Electronics, Computer Science and Technology (ICERECT)*, pages 34–38, 2015.
- [4] Aidan Boyd, Adam Czajka, and Kevin Bowyer. Deep learning-based feature extraction in iris recognition: Use existing models, fine-tune or train from scratch? Number: arXiv:2002.08916.
- [5] Aidan Boyd, Shivangi Yadav, Thomas Swearingen, Andrey Kuehlkamp, Mateusz Trokielewicz, Eric Benjamin, Piotr Maciejewicz, Dennis Chute, Arun Ross, Patrick Flynn, Kevin Bowyer, and Adam Czajka. Post-mortem iris recognition—a survey and assessment of the state of the art. 8:136570–136593.
- [6] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, Nov 1986.
- [7] John G. Daugman. High confidence visual recognition of persons by a test of statistical independence. 15(11).

- [8] Guoyang Liu, Weidong Zhou, Lan Tian, Wei Liu, Yingjian Liu, and Hanwen Xu. An efficient and accurate iris recognition algorithm based on a novel condensed 2-ch deep convolutional neural network. 21(11):3721.
- [9] Jus Lozej, Blaz Meden, Vitomir Struc, and Peter Peer. End-to-end iris segmentation using u-net. In *2018 IEEE International Work Conference on Bioinspired Intelligence (IWOBI)*, pages 1–6. IEEE.
- [10] Jasem Rahman Malgheet, Noridayu Bt Manshor, and Lilly Suriani Affendey. Iris recognition development techniques: A comprehensive review. 2021:1–32.
- [11] M. De Marsico, Alfredo Petrosino, and Stefano Ricciardi. Iris recognition through machine learning techniques: A survey. *Pattern Recognit. Lett.*, 82:106–115, 2016.
- [12] Shervin Minaee and Amirali Abdolrashidi. DeepIris: Iris recognition using a deep learning approach. Number: arXiv:1907.09380.
- [13] Shervin Minaee, Amirali Abdolrashidi, and Yao Wang. An experimental study of deep convolutional features for iris recognition. Number: arXiv:1702.01334.
- [14] N. Susitha and Ravi Subban. Reliable pupil detection and iris segmentation algorithm based on sps. *Cogn. Syst. Res.*, 57(C):78–84, oct 2019.
- [15] Maciej Szymkowski, Piotr Jasiński, and Khalid Saeed. Iris-based human identity recognition with machine learning methods and discrete fast fourier transform. 17(3):309–317.