

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 — 9 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, 9 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 compara la capacidad de reconocimiento de individuos que tienen las redes neuronales. Para ello, utilizando *fine-tuning*, adaptaremos cuatro variaciones de nuestro dataset a la red VGG16. Estas variaciones son por la utilización de *data augmentation*, tanto a imágenes del ojo sin pre-procesar como a imágenes del ojo donde se ha normalizado el iris.

Los resultado muestra que la red neuronal ha funcionado de forma más eficiente en las imágenes donde no se había aislado la zona del iris. Así mismo, las 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

A **brief** presentation of the topic addressed in the project.

Keywords

keywords separated by commas.

Índice general

Índice general	iii
Índice de figuras	v
Índice de tablas	vi
Memoria	1
1. Introducción	3
1.1. Outline	4
2. Objetivos del proyecto	5
3. Conceptos teóricos	7
3.1. Biometría	7
3.2. Inteligencia Artificial	8
4. Técnicas y herramientas	15
4.1. Hardware	15
4.2. Dataset	15
4.3. Github	15
4.4. Python	15
4.5. Scrum	16
5. Aspectos relevantes del desarrollo del proyecto	17
5.1. Optimización de los notebooks previos	17
5.2. <i>data augmentation</i>	18

5.3. Clasificación de las imágenes	19
6. Trabajos relacionados	21
7. Conclusiones y Líneas de trabajo futuras	25
7.1. Conclusiones	25
7.2. Líneas de trabajo futuras	25
 Apéndices	 26
Apéndice A Plan de Proyecto Software	29
A.1. Introducción	29
A.2. Planificación temporal	29
A.3. Estudio de viabilidad	31
 Apéndice B Especificación de Requisitos	 33
B.1. Introducción	33
B.2. Objetivos generales	33
B.3. Catalogo de requisitos	33
B.4. Especificación de requisitos	33
 Apéndice C Especificación de diseño	 35
C.1. Introducción	35
C.2. Diseño de datos	35
C.3. Diseño procedimental	35
C.4. Diseño arquitectónico	35
 Apéndice D Documentación técnica de programación	 37
D.1. Introducción	37
D.2. Estructura de directorios	37
D.3. Manual del programador	37
D.4. Compilación, instalación y ejecución del proyecto	37
D.5. Pruebas del sistema	37
 Apéndice E Documentación de usuario	 39
E.1. Introducción	39
E.2. Requisitos de usuarios	39
E.3. Instalación	39
E.4. Manual del usuario	39
 Bibliografía	 41

Índice de figuras

3.1. Eliminación del reflejo de la pupila.	8
3.2. Ejemplo de <i>data augmentation</i> por ruido gaussiano.	9
3.3. Ejemplo de transformacion de identidad.	10
3.4. Ejemplo de transformacion por reflexión.	11
3.5. Ejemplo de transformacion por escalamiento.	11
3.6. Ejemplo de transformacion por traslación.	12
3.7. Ejemplo de transformacion por rotación.	12
5.8. Workflow utilizado	18

Índice de tablas

3.1. Transformaciones afines aplicadas en el trabajo.	10
---	----

Memoria

Introducción

El *Oxford Learners Dictionaries* ¹ define la biometría como la utilización de características humanas parar 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 [5].

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 pre-entrenada ImageNet², a la que, para que los resultados de clasificación de la red neuronal se adapten a nuestro *dataset* de imágenes, se le ha aplicado *fine-tuning* ³.

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

²ImageNet es un proyecto donde se proporciona una gran base de datos de imágenes para usos no comerciales <https://www.image-net.org/>

³El *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.

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*.

Así mismo, en ambos enfoques, se han utilizado técnicas de *data augmentation* ⁴ 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.

⁴El 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 [5], 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 [4]. 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 [4].

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 3.1 ⁵.

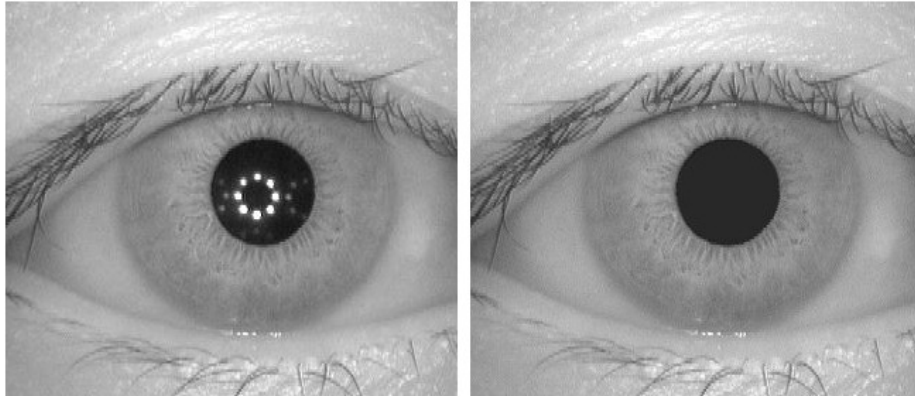


Figura 3.1: Eliminación del reflejo de la pupila [2].

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 ⁶.

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. ⁷

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

⁵El proceso de la toma de muestras se describe en <http://www.cbsr.ia.ac.cn/IrisDatabase.htm>

⁶En base a la definición de <https://dle.rae.es/inteligencia>.

⁷En base a la definición de <https://www.ibm.com/cloud/learn/deep-learning>.

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 ⁸.

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 3.2.

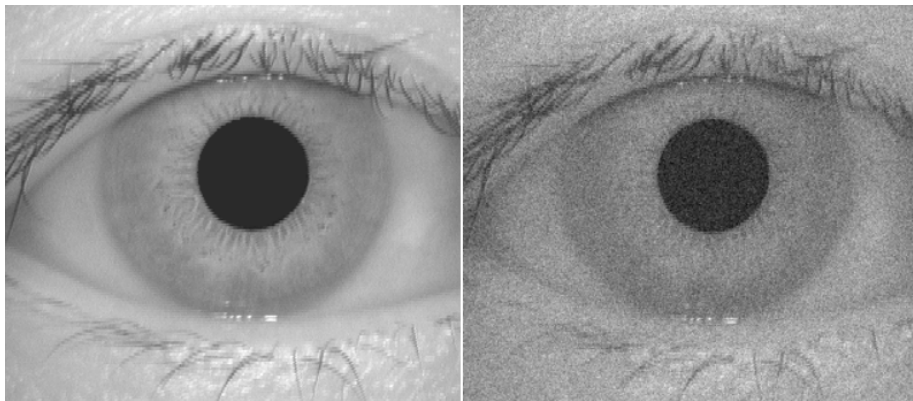


Figura 3.2: Ejemplo de *data augmentation* por ruido gaussiano.

⁸Por ejemplo, si se rota una imagen 2 grados, la diferencia con la original será mucho menor que si se rota 180 grados.

Transformaciones afines				
Identidad	Reflexión	Escalamiento	Traslación	Rotación
Figura 3.2	Figura 3.2	Figura 3.2	Figura 3.2	Figura 3.2

Tabla 3.1: Transformaciones afines aplicadas en el trabajo.

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 3.2.

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.

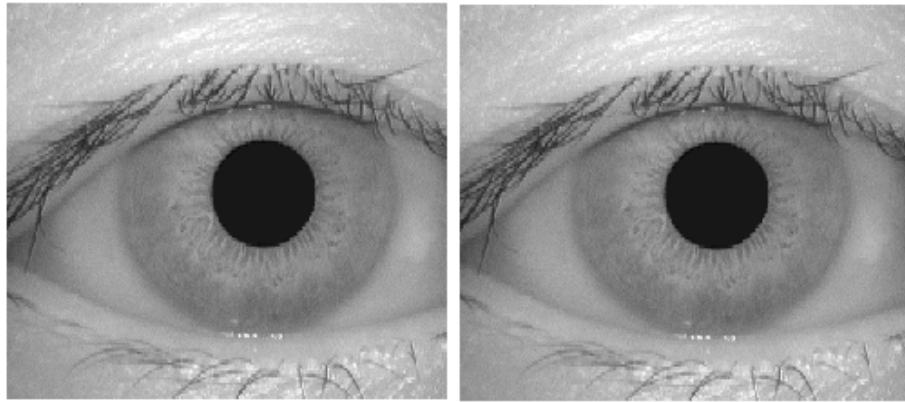


Figura 3.3: Ejemplo de transformacion de identidad.

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

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

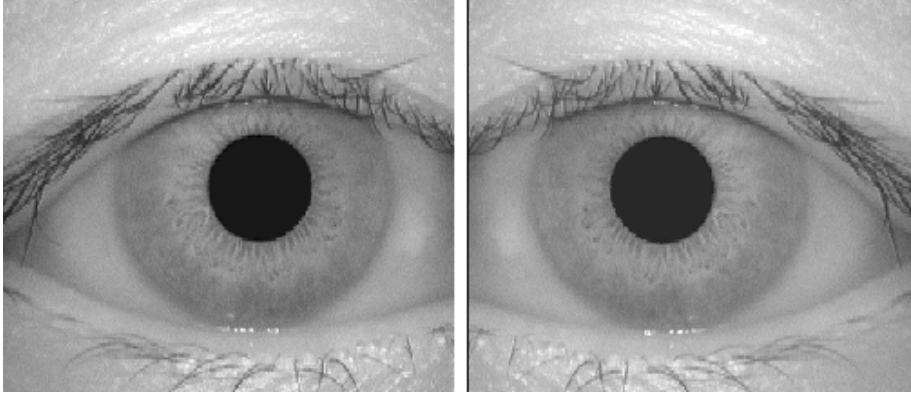


Figura 3.4: Ejemplo de transformacion por reflexión.

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

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

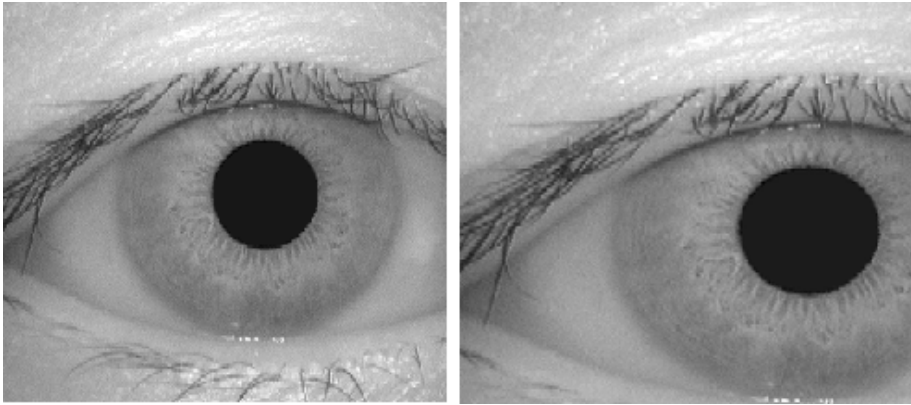


Figura 3.5: Ejemplo de transformacion por escalamiento.

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

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

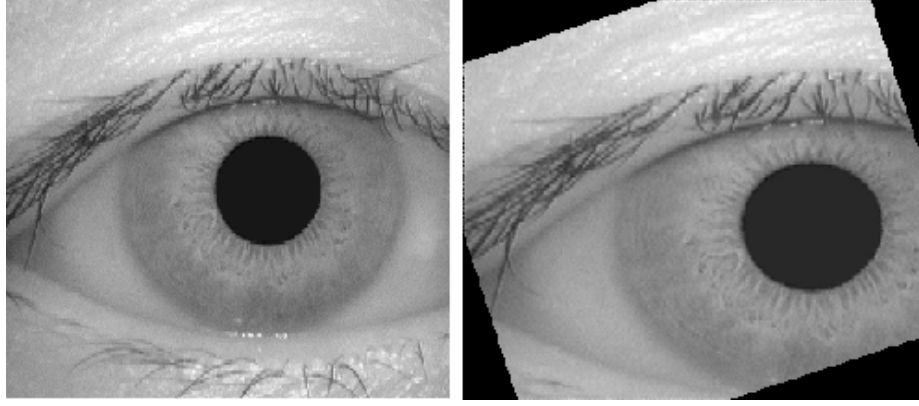


Figura 3.6: Ejemplo de transformacion por traslación.

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

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

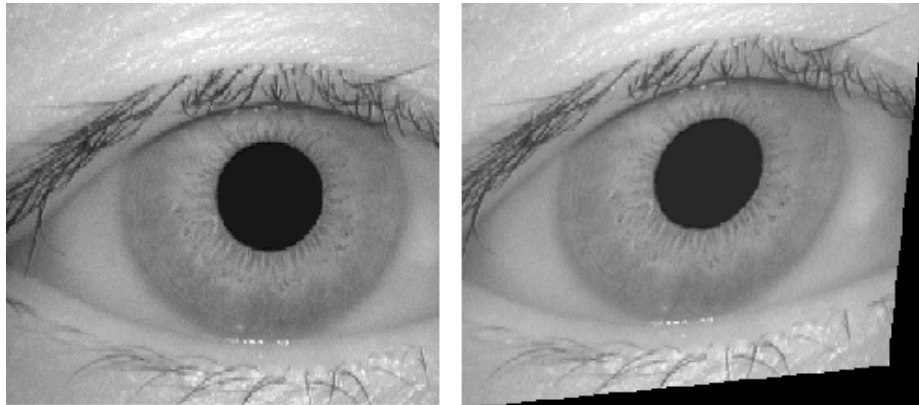


Figura 3.7: Ejemplo de transformacion por rotación.

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

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 (referencia), la parte del iris es la que permite identificar a las personas.

Segmentación

Detección de las distintas zonas del ojo.

Normalización

Normalización del iris y proyección en coordenadas polares.

Fine-tuning

El *fine-tuning* es una técnica que se utiliza para poder ajustar los parámetros de las redes neuronales pre-entrenadas a los necesarios en relación con el dataset con el que se trabaja.

Técnicas y herramientas

Explicación de la sección

4.1. Hardware

Para la ejecución de los notebooks de este proyecto se ha contado con una máquina de la UBU de x e y características.

Se ha utilizado conexión ssh con ella y putty para la modificación de los notebooks a través de jupyter notebook.

4.2. Dataset

Se ha utilizado el dataset de CASIA para la imágenes de ojos.

4.3. Github

Para el control de versiones y el seguimiento de fases del proyecto

4.4. Python

Además se ha utilizado python para todo el proyecto, esto quiere decir pre-procesado, creación y utilización de redes neuronales y clasificación contécnicas de machine learning.

Entre las principales librerías utilizadas se encuentran:

- os, para el acceso a los directorios
- numpy, para trabajar con las imágenes a nivel de arrays
- scikit-image, para la transformación de las imágenes y el uso de dataset
- tensorflow, para la modificación de las redes neuronales
- keras, para el manejo de las redes neuronales
- matplotlib, para las gráficas

Redes neuronales pre-entrenadas

Así mismo, se ha utilizado la red neuronal pre-entrenada, basada en U-Net y accesible desde <https://github.com/jus390/U-net-Iris-segmentation>, la cual ya había sido entrenada para la segmentación del iris.

Finalmente, para el último proceso del proyecto, se utiliza imagenet como red neuronal pre-entrenada central.

Visual Studio

Tanto para la redacción de la memoria con latex como para el la creación de los notebooks.

4.5. 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.

Aspectos relevantes del desarrollo del proyecto

El proyecto se divide en una fase preliminar, consistente en la adecuación del código previo, correspondiente en el TFG el cual se ha hecho extensible en este trabajo y tres propuestas diferentes para la identificación biomédica de personas a través de una imagen de su ojo utilizando el iris como parte central de este reconocimiento.

5.1. Optimización de los notebooks previos

La primera fase trata de la optimización del código previo y la creación de un pipeline con una configuración que permitiese un mejor control de las funciones. Se han creado funciones correspondientes a cada una de las fases y se han encapsulado en forma de partes de una pipeline, con el único argumento de entrada de la configuración.

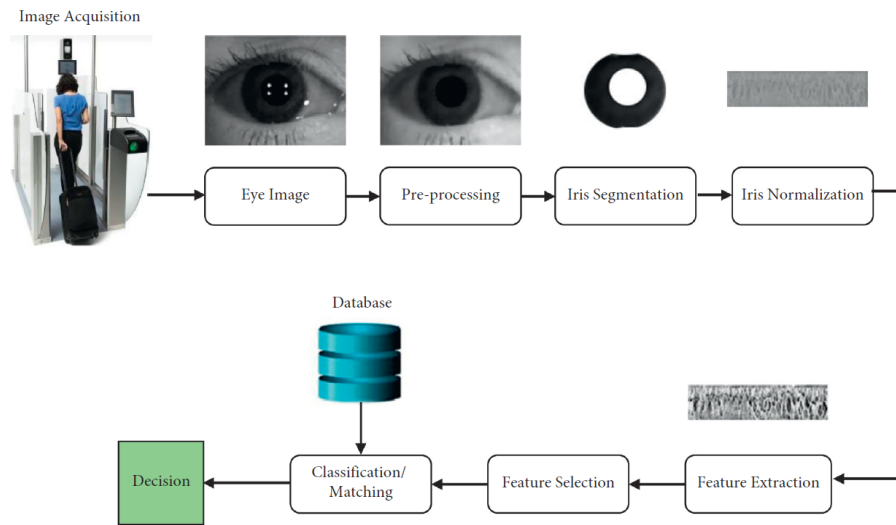


Figura 5.8: Ejemplo de workflow del paper [5]

Pipeline y configuración

Así pues, esta configuración permite establecer cuales van a ser los parámetros de la pipeline, principalmente (aunque puede ser fácilmente extensible), la ubicación del directorio del dataset, el tamaño del dataset de entrenamiento para la creación de los modelos de redes neuronales o los valores de data augmentation que se aplican.

Si se desea correr con raw, se deberá de establecer raw en la configuración del pipeline 4.

5.2. *data augmentation*

También del data augmentation, en el cual se aplica tanto ruido gaussiano (de 2.5, 5 y 7.5) como transformaciones afines (mat reflexion, mat scale, rotación y shear), siendo realizadas de forma aleatoria, con lo cual los supuestos pueden ser:

- Imagen sin data augmentation
- Imagen con ruido gaussiano
- Imagen con transformaciones afines y

- Imagen con ruido gaussiano y transformaciones afines. Este dataset será el base para todo el proyecto.

Las fases del pipeline anterior, y por lo tanto las funciones creadas en el pipeline son: 1.Tratar el dataset, de forma que las imágenes, distribuidas en distintos directorios, se guardan en uno solo. Posteriormente, se realiza en mezclado de las imágenes, el dataset se divide en dos partes, una que será utilizada en el proceso (70 por ciento) y, finalmente, se guarda una parte del dataset que será utilizada para obtener los resultados finales.

La segunda fase se trata de la elección de la mejor forma de clasificar las imágenes de entrada:

5.3. Clasificación de las imágenes

	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

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.

Preprocesamiento y Machine Learning

La propuesta en el TFG anterior, en la cual se realiza un preprocesamiento de las imágenes del dataset original. Este preprocesamiento consiste primer

lugar en la segmentación de las imágenes del iris con una red neuronal pre-entrenada precisamente para realizar esta acción. En segundo lugar, se realiza una extracción del iris a través de una binarización de las partes del ojo y una extracción del iris, a la que se le aplica una normalización para que quede proyectado. La siguiente fase de esta primera propuesta es la extracción de características (quitándole las dos últimas capas a una red neuronal) de la imagen normalizada para posteriormente utilizar una red neuronal pre-entrenada con imagenet (de hecho 3 redes de la cual se elige la mejor). Posteriormente, estas características extraídas se pasan a modelos de ML, que son los que realizarán la clasificación.

Preprocesamiento y Deep Learning

La segunda propuesta es la utilización de una red neuronal de VGG16, pre-entrenada con imagenet, a la que se le aplica un *fine-tuning* del dataset de CASIAV1 para que sea capaz de clasificar las imágenes de acuerdo a las clases establecidas en el dataset.

Deep Learning

La tercera propuesta se basa en la utilización del dataset, directo sin preprocesamiento, para la fase del *fine-tuning* y su posterior clasificación (para la clasificación de imágenes en las fases que han sufrido preprocesamiento, le deberían de pasar imágenes que primero hayan entrado al pipeline hasta la fase de normalización!!). (temporal) Los resultados arrojan un acierto mayor en el caso del pre-proceso + red neuronal, por lo que se puede concluir que, en base a los parámetros establecidos en este estudio, la utilización de una fase de preprocesamiento que elimine las partes que los estudios determinan no tan útiles para la clasificación humana (vease, todo aquello que no es parte del iris) y posteriormente utiliza el *fine-tuning* para adecuar una red neuronal pre-entrenada a las clases de las que se compone el dataset.

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 [6] 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 [7], 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 [5] 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 cualidad 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.

De Marsico et al. [44] utiliza también el dataset de casia V! (parece que también Susitha and Subban [81]) para medir el accuracy y Lozej et al. [176] junto con Unet para El iris segmentation.

(esto casi se podría quitar pues no es tanto el foco de nuestro trabajo) Varkarakis et al. [179] también utiliza una cnn para segmentar el iris

Bakshi et al. [72] utiliza filtro gaussiano combinado con Hough detección de líneas pero nadie utiliza solo gaussiano y/o transformaciones afines (al menos en este recopilatorio).

[3] 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. (It is better to take the best-performing model trained on either general-purpose or face images and fine-tune it to iris recognition task, rather than train own network) from scratch.

Existen varios papers que utilizan fully CNN, y lo mismo para feature extraction.

.... (utilizando Machine learning y deep learning)

Y para el tema de clasificación de personas con el ojo, se ha encontrado ...

Por otro lado, en temas de *fine-tuning* con redes neuronales, se tiene ...

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 determinan 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 gausseano 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] 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.
- [4] 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.
- [5] Jasem Rahman Malgheet, Noridayu Bt Manshor, and Lilly Suriani Affendey. Iris recognition development techniques: A comprehensive review. 2021:1–32.
- [6] Shervin Minaee and Amirali Abdolrashidi. DeepIris: Iris recognition using a deep learning approach. Number: arXiv:1907.09380.
- [7] Shervin Minaee, Amirali Abdolrashidi, and Yao Wang. An experimental study of deep convolutional features for iris recognition. Number: arXiv:1702.01334.