

Facultad de Informática

Grado en Ingeniería de Computadores



TRABAJO DE FIN DE GRADO

Desarrollo de una aplicación para caracterización de rocas mediante
Inteligencia Artificial

Development of an application for rock characterization using Artificial
Intelligence

Sara Núñez Sánchez y Luis Ruiz de Nicolas

Directores:

José Ignacio Hidalgo Pérez

Sol López Andrés

Junio 2021

Curso académico 2020-2021

Tabla de contenido

Resumen	5
Abstract	6
1. Introducción	7
1. Introduction	15
2. Sistemas de Clasificación de Imágenes basados en Inteligencia Artificial	23
2.1. Introducción a la IA	23
2.2. ¿Qué es una red neuronal?	24
2.3. Principales tipos de redes neuronales	25
2.3.1. Clasificación de redes neuronales según la topología de la red.	25
2.3.2. Clasificación de redes según el método de aprendizaje.	25
2.4. Redes convolucionales	27
2.5. Red AlexNet	27
2.5.1. Capas convolucionales	28
2.5.2. Capas “fully-connected”	28
2.5.3. Función ReLu	29
2.5.4. Función Softmax	29
2.5.5. Data Augmentation	30
2.5.6. Pooling	30
2.5.7. Técnica DropOut	31
2.6. Deep Learning	31
2.7. Detección, clasificación y segmentación	32
2.8. Framework Nvidia Digits y Pytorch.	32
3. Metodología de trabajo y desarrollo del proyecto	35
3.1. Desarrollo del Modelo de red neuronal.	35
3.2. Diseño e investigación	35
3.3. Obtención del conjunto de datos	35
3.3. Normalización y clasificación	36
3.5. Implementación de la transferencia de aprendizaje	37
3.6. Entrenamiento	38

3.7. Métricas de evaluación y análisis de los resultados	38
3.8. Generación del modelo NN final	40
3.9. Rendimiento del entrenamiento en distintos dispositivos y con diferentes frameworks	41
4. Desarrollo aplicación web GUI con Flask: 'StoNet'	43
4.1. Despliegue del modelo	44
4.2. Depuración del uso y funcionamiento de la aplicación.	44
4.3 Despliegue sobre un servidor. (Pendiente de completar)	45
5. Resultados experimentales.	46
5.1 Entrenamiento de la red	46
5.2 Prueba de clasificación de imágenes no utilizadas en el entrenamiento	49
6. Conclusiones	52
6.1. Retos a los que nos hemos tenido que enfrentar.	52
6.2. Aportación de cada uno de los miembros	54
6.Conclusions	58
6.1. Challenges we have had to face	58
7.Bibliografía	59

Resumen

El problema de la identificación y caracterización de las rocas que llevan a cabo los colectivos que trabajan con muestras geológicas como son los arquitectos, arqueólogos, historiadores etc. ha sido siempre un problema de importancia y de gran dificultad. Para el reconocimiento de minerales y rocas existen distintas técnicas que son aplicadas en diversos campos y que tienen diferentes tasas de éxito. El tipo de análisis que se lleva a cabo depende en gran medida de la posibilidad de llevar la muestra a un laboratorio o si por el contrario, la identificación debe realizarse in situ. Una de las técnicas más utilizada es la que está basada en la identificación mediante microscopía óptica. Para ello se precisa un microscopio petrográfico, y a partir de imágenes de láminas delgadas de las rocas, es posible determinar la composición mineralógica y así clasificar el tipo de roca. En el caso de las rocas utilizadas en la construcción de utensilios de uso común antiguos, como son las piedras de afilar, es importante determinar la hipotética cantera de origen. Para ello, es necesario el uso de otras técnicas complementarias como son la difracción de rayos X (DRX) y la fluorescencia de rayos X (FRX), entre otras. Este trabajo de fin de grado tiene como principal objetivo la creación de una herramienta informática de identificación de rocas, basada en un sistema de inteligencia artificial y asequible para colectivos no geológicos. El sistema es capaz de, a partir de imágenes obtenidas con un microscopio petrográfico de diferentes canteras, asignarlas una posible cantera origen entre las que previamente han sido analizadas. Se trata de un sistema que permite ser usado tanto en dispositivos móviles, tablets y smartphones como en ordenadores. Para ello se ha recopilado un conjunto de muestras de estas canteras 'históricas' y de piedras de afilar con las que han sido entrenados los algoritmos de clasificación del sistema basados en redes neuronales.

Palabras clave: Inteligencia artificial, modelos, datos, características, desarrollo, entrenamiento, canteras, areniscas, piedras de afilar, microscopio petrográfico

Abstract

The problem of identification and characterization of rocks carried out by groups working with geological samples such as architects, archaeologists, historians, etc. has always been an important and very difficult problem. For the recognition of minerals and rocks there are different techniques that are applied in various fields and that have different success rates. The type of analysis that is carried out depends to a great extent on the possibility of taking the sample to a laboratory or if, on the contrary, the identification must be done in situ. One of the most used techniques is based on identification by optical microscopy. This requires a petrographic microscope, and from images of thin slices of the rocks, it is possible to determine the mineralogical composition and thus classify the type of rock. In the case of rocks used in the construction of commonly used ancient tools, such as whetstones, it is important to determine the hypothetical quarry of origin. For this, it is necessary to use other complementary techniques such as X-ray diffraction (XRD) and X-ray fluorescence (XRF), among others. The main objective of this final degree project is the creation of a computer tool for rocks identification, based on an artificial intelligence system and accessible to non-geological groups. The system is able, from images obtained with a petrographic microscope of different quarries, to assign them a possible quarry origin among those that have been previously analyzed. It is a system that can be used on mobile devices, tablets and smartphones as well as computers. For this purpose, a set of samples of these 'historical' quarries and sharpening stones have been collected and used to train the classification algorithms of the system based on neural networks.

Key Words: Artificial intelligence, models, data, features, development, training, quarries, sandstones, whetstones, petrographic microscope.

1. Introducción

La identificación de rocas para colectivos que trabajan con muestras geológicas, como arquitectos, arqueólogos, historiadores, etc. es un problema de máxima importancia y de difícil solución, No obstante, existen diferentes técnicas de reconocimiento de minerales y rocas que se aplican en campos muy diversos y que tienen distintas tasas de éxito. El tipo de análisis a realizar dependerá de si la muestra se puede o no llevar al laboratorio.

En materiales de patrimonio arqueológico es fundamental la utilización de técnicas *no invasivas* y/o *micro invasivas*, ya que uno de los aspectos más importantes a tener en cuenta es la conservación de la muestra. Sólo en algunos casos cuando la cantidad de muestra lo permita se utilizarán técnicas *invasivas*.

También puede ser necesario utilizar técnicas *destructivas*, que requieren destruir la muestra para poder realizar el análisis, o si permite sólo destruirla parcialmente, se podrían utilizar técnicas *no destructivas*, que, aunque la muestra no vuelva a su estado original, permita ser estudiada y analizada mediante otras técnicas posteriormente.

Las técnicas de reconocimiento de minerales y rocas más utilizadas son el *reconocimiento microscópico*, que tratan de observar y estimar las proporciones de los minerales más abundantes y las relaciones texturales entre ellos. Dentro de éstas se encuentra la microscopía óptica. En ella, utilizando un microscopio petrográfico o metalogénico, a partir de láminas delgadas o probetas pulidas, respectivamente, es posible determinar la composición mineralógica y clasificar el tipo de roca. Para determinar la hipotética cantera origen procedencia geológica y geográfica de las rocas se necesita de otras técnicas complementarias como son la difracción de rayos X (DRX) y la fluorescencia de rayos X (FRX), entre otras. En los trabajos arqueológicos y de patrimonio, estas técnicas se utilizan para identificar las canteras de origen con los objetos de estudio.

La utilización de estas técnicas permite conocer la composición química y mineralógica, la morfología, la textura, etc., de los objetos arqueológicos para, por ejemplo, poder compararlas con las posibles canteras de origen, así como a los objetos arqueológicos entre sí.

Este trabajo de fin de grado tiene como objetivo ofrecer una herramienta de identificación de determinadas rocas, asequible para colectivos no geológicos, basado en un sistema de inteligencia artificial, capaz de, a partir de imágenes

microscópicas de objetos de uso común, como son las piedras de afilar, asignar una posible cantera de origen de las que previamente hayan sido analizadas. Se trata de un sistema que se puede usar tanto en ordenadores como tablets o dispositivos móviles.

Las muestras utilizadas en este proyecto se tomaron de diferentes canteras “históricas” ibéricas y griegas (**Figura 1** y **Figura 2**) y de materiales de uso común realizados con piedras supuestamente del mismo tipo, como son las piedras de afilar, que se utilizarán para entrenar los algoritmos de clasificación.



FIGURA 1 CANTERA DE CUARCITA DE ELOUNDA (GRECIA). FOTO CORTESÍA DE PEDRO MOYA-MALENO



FIGURA 2 CANTERA DE ARENISCA DE LOS MOLARES (CIUDAD REAL). FOTO CORTESÍA DE PEDRO MOYA-MALENO

El modelo de inteligencia artificial utilizado es de los conocidos como de redes neuronales artificiales (neural networks).

La utilización de la herramienta descrita permitirá la localización de canteras para muestras en lámina delgada sin la participación directa de profesionales de la geología. Aunque estos seguirán siendo necesarios para la identificación y descripción de las láminas delgadas. Esta idea surge a raíz de un estudio llevado a cabo sobre las piedras de afilar y el posible origen de éstas [1.]

Metodología

La metodología propuesta en este trabajo tiene varias etapas:

- Primera etapa: Obtener muestras de rocas de diferentes canteras y piedras de afilar u objetos de uso común fabricados con piedras como son las piedras de molino, manos de mortero, etc.
- Segunda etapa: Realización de las láminas delgadas de las muestras.
- Tercera etapa: Adquisición de las fotografías en el microscopio petrográfico con diferentes condiciones de trabajo como son: aumentos, polarizadores, extensión de salida, etc.
- Cuarta etapa: Entrenamiento de una red neuronal artificial que, una vez entrenada, sea capaz de clasificar nuevas imágenes, como perteneciente

a una de las diferentes canteras de las que se han obtenido las imágenes obtenidas en la etapa anterior.

- Quinta etapa: Desarrollo de la infraestructura web que permite la consulta del modelo.

El trabajo de la primera y segunda etapa estaba previamente realizado y pendiente de publicación en “Hacia una caracterización de las piedras de afilar naturales y sus canteras fuente históricas en la Península Ibérica” por S. López-Andrés, V. López-Acevedo Cornejo, X. Arroyo Rey y Pedro P.R. Moya-Maleno en el libro “Al otro lado del filo: armas, herramientas y piedras de afilar desde la prehistoria”. “Hacia una caracterización de las piedras de afilar naturales y sus canteras fuente históricas en la Península Ibérica” en (en prensa)). [2.]

Para la tercera etapa se han utilizado las láminas delgadas de ocho canteras “históricas” de la Península ibérica y de Grecia (**Figura 3 y Figura 4**). A partir de estas láminas delgadas, se han realizado una serie de imágenes con un microscopio petrográfico Zeiss Primotech, dotado con objetivos de 5x, 10x y 20x para analizar dichas muestras, y en el que se pueden acoplar uno o dos polarizadores. De cada lámina delgada se ha realizado un barrido completo de la misma obteniendo en torno a 2000 imágenes. Para el entrenamiento de la red, se han utilizado las imágenes de 5 y 20 aumentos y utilizando un solo polarizador.



FIGURA 3 MUESTRAS DE LÁMINAS DELGADAS UTILIZADAS



FIGURA 4 MICROSCOPIO PETROGRÁFICO ZEISS MODELO PRIMOTECH. UNIDAD DE TÉCNICAS GEOLÓGICAS. CAI DE CIENCIAS DE LA TIERRA Y ARQUEOMETRÍA. UCM.

Una vez se dispone de todo el conjunto de imágenes, se trata de encontrar una red neuronal que, una vez entrenada, de buenos resultados para la clasificación de éstas. En este proyecto se han entrenado varias redes neuronales, eligiendo finalmente dos de ellas, con las cuales se ha obtenido un mejor resultado para este trabajo.

Para poder entrenar la red, se la dota con un dataset, que es el conjunto de todas las imágenes obtenidas y el cual tiene que ser lo suficientemente amplio, para que cuando observe una imagen desconocida, tenga la suficiente información para poder clasificarla dentro de una de las clases establecidas.

En este caso, será una imagen de una roca, y la clasificará en una de las canteras que se le han proporcionado, indicando la probabilidad que tiene de que sea de cada una de ellas.

Objetivos

El objetivo principal de este trabajo es el **desarrollo e implementación de un sistema de clasificación de rocas mediante redes neuronales**. A su vez, éste se materializa en un objetivo secundario de este trabajo que es: el desarrollo de una aplicación informática, que permita al usuario interactuar con la misma y obtener clasificación de las rocas partiendo de una imagen microscópica de una manera sencilla y amigable

Para que sea de utilidad, esta aplicación debe permitir que un usuario pueda subir una imagen y obtener el resultado de la clasificación de esta imagen en un tiempo aceptable y con la mayor fiabilidad posible. El programa tendrá cargados varios modelos, según se quiera clasificar una imagen con unos u otros aumentos y dependiendo del polarizador utilizado. Además, en el desarrollo de la aplicación se plantea que, dentro de esta aplicación, existirán dos versiones, una en la que no es necesario que el usuario se registre y otra en la que podrá hacerlo y, por tanto, tendrá más funcionalidades disponibles.

En la versión sin registro, el usuario sólo puede elegir una de las redes entrenadas, aunque podrá elegir diferentes aumentos. En esta versión sólo se permitirá subir una imagen, que será clasificada y se mostrará en la aplicación las probabilidades de que pertenezca a las diferentes canteras. El objetivo es que el usuario pueda probar la funcionalidad y utilidad del sistema sin necesidad de hacer un proceso de registro

Sin embargo, en la versión con registro, el usuario podrá elegir entre las dos redes entrenadas, y también diferentes modelos, según los aumentos y el polarizador utilizado. Otra de las ventajas de estar registrado, es que se podrán subir varias fotos simultáneamente, que el programa clasificará a la vez, obteniendo de cada una de ellas, de forma simultánea, las probabilidades de pertenecer a las diferentes canteras. Este programa también permitirá descargar un documento donde aparezcan todos los resultados.

Como hemos dicho, el objetivo fundamental es que sea un programa intuitivo y sencillo, donde el usuario sólo tenga que subir una imagen, obteniendo fácilmente los resultados de su clasificación. Finalmente, la aplicación debe ser ampliable para futuras investigaciones, es decir debe dejar abierta la incorporación de nuevas canteras y el consiguiente (re) entrenamiento de las redes

Canteras utilizadas

Las muestras que se han utilizado en este proyecto se han extraído de diferentes canteras históricas que se recogen en la siguiente **(Tabla 1)**, junto a se muestra la clasificación petrográfica y la localización de dichas canteras las mismas. De las canteras elegidas, siete de ellas son de rocas sedimentarias tipo arenisca (subarcosa y arcosa) y la octava de una roca metamórfica, cuarcita. Las areniscas son rocas sedimentarias detríticas, que contienen al menos, el 50% de granos del tamaño de arena y están compuestas por cuarzo y feldspatos, como minerales fundamentales. Dentro de las areniscas hay varios tipos, de los cuales, dos de ellos se han identificado en estas canteras, y son arcosas y subarcosas, que se diferencian en la proporción de los minerales fundamentales. Las cuarcitas son rocas metamórficas formadas fundamentalmente por cuarzo.

Canteas	Localización	Tipo
Cantera 1	El Polvorín (Toledo)	Subarcosa
Cantera 2	San Antonio (Toledo)	Subarcosa
Cantera 3	Los Molares (Ciudad Real)	Subarcosa
Cantera 4	Pozo paturro (Ciudad Real)	Arcosa
Cantera 5	Arroyo Santo (Jaén)	Subarcosa
Cantera 6	Los Huertos (Soria)	Arcosa
Cantera 7	El Soto (Soria)	Arcosa
Cantera 8	Elounda (Creta, Grecia)	Cuarcita

TABLA 1 CANTERAS ANALIZADAS Y TIPOS DE ROCA QUE LAS FORMAN

Organización de la memoria

El resto de la memoria está organizada de la siguiente manera:

- En el capítulo 2 se realiza un somero recorrido por el mundo de la inteligencia artificial y los distintos campos que los forman explicando y detallando cada uno de ellos.
- El capítulo 3 está compuesto por una explicación del proceso de desarrollo del modelo de red neuronal que ha sido utilizado en el proyecto.
- En el capítulo 4 se cuenta cuál ha sido el framework elegido para dar una interfaz a la aplicación y así poderla desplegar en una página web.
- En el capítulo 5 se pueden observar los resultados experimentales que han sido obtenidos con el uso de la aplicación
- En el capítulo 6 se exponen los retos a los que nos hemos tenido que enfrentar, así como las conclusiones obtenidas una vez realizado el proyecto y por último el reparto del trabajo entre cada uno de los integrantes del grupo.
- En el capítulo 7 hay un resumen de la bibliografía utilizada a lo largo de todo el documento.

1. Introduction

Rock identification for groups working with geological samples, such as architects, archaeologists, historians, etc. is a problem of utmost importance and difficult to solve. However, there are different techniques for mineral and rock recognition that are applied in very diverse fields and have different success rates. The type of analysis to be performed will depend on whether or not the sample can be taken to the laboratory.

In archaeological heritage materials it is essential to use non-invasive and/or micro-invasive techniques, since one of the most important aspects to take into account is the preservation of the sample. Invasive techniques will be used only in some cases when the amount of sample allows it.

It may also be necessary to use destructive techniques, which require destroying the sample in order to perform the analysis, or if it allows only partial destruction, non-destructive techniques could be used, which, although the sample does not return to its original state, allow it to be studied and analyzed by other techniques at a later date.

The most commonly used mineral and rock recognition techniques are microscopic recognition, which try to observe and estimate the proportions of the most abundant minerals and the textural relationships between them. Among these is optical microscopy. In it, using a petrographic or metallogenic microscope, from thin slices or polished specimens, respectively, it is possible to determine the mineralogical composition and classify the type of rock. To determine the hypothetical quarry geological and geographical origin of the rocks, other complementary techniques such as X-ray diffraction (XRD) and X-ray fluorescence (XRF), among others, are needed. In archaeological and heritage works, these techniques are used to identify the quarries of origin with the objects of study.

The use of these techniques allows us to know the chemical and mineralogical composition, morphology, texture, etc., of the archaeological objects in order, for example, to compare them with the possible quarries of origin, as well as the archaeological objects with each other.

The aim of this final degree project is to offer a tool for the identification of certain rocks, accessible to non-geological groups, based on an artificial intelligence system, capable of assigning, from microscopic images of commonly used objects, such as whetstones, a possible quarry of origin from those that have

been previously analyzed. It is a system that can be used on computers, tablets or mobile devices.

The samples used in this project were taken from different "historical" Iberian and Greek quarries (**Figure 1** and **Figure 2**) and from commonly used materials made with stones supposedly of the same type, such as whetstones, which will be used to train the classification algorithms.



FIGURE 1 QUARTZITE QUARRY OF ELOUNDA (GREECE). PHOTO COURTESY OF PEDRO MOYA-MALENO



FIGURE 2 LOS MOLARES SANDSTONE QUARRY (CIUDAD REAL). PHOTO COURTESY OF PEDRO MOYA-MALENO

The artificial intelligence model used is one of those known as artificial neural networks. The use of the described tool will allow the location of quarries for thin film samples without the direct participation of professional geologists. Although they will still be necessary for the identification and description of thin films. This idea arises from a study carried out on sharpening stones and their possible origin [Bibliography: study López-Andrés et al. 2022 (in press)].

Methodology

The methodology proposed in this work has several stages:

First stage: Obtaining rock samples from different quarries and sharpening stones or objects of common use manufactured with stones such as millstones, mortar hands, etc.

Second stage: Making of the thin sheets of the samples.

Third stage: Acquisition of photographs in the petrographic microscope with different working conditions such as: magnifications, polarizers, output extension, etc.

Fourth stage: Training of an artificial neural network that, once trained, is able to classify new images as belonging to one of the different quarries from which the images obtained in the previous stage have been obtained.

Fifth stage: Development of the web infrastructure that allows the consultation of the model.

The work of the first and second stages was previously carried out and pending publication in "Towards a characterization of natural whetstones and their historical source quarries in the Iberian Peninsula" by the CAI Geológicas (S. López-Andrés, V. López-Acevedo Cornejo, X. Arroyo Rey and Pedro P.R. Moya-Maleno in the book "Al otro lado del filo: armas, herramientas y piedras de afilar desde la prehistoria". "Towards a characterization of natural whetstones and their historical source quarries in the Iberian Peninsula" in (in press)).

For the third stage, thin slices from eight "historical" quarries in the Iberian Peninsula and Greece have been used (**Figure 3** and **Figure 4**). From these thin films, a series of images were taken with a Zeiss Primotech petrographic microscope, equipped with 5x, 10x and 20x objectives to analyze these samples, and to which one or two polarizers can be attached. A complete scan of each thin

film was performed, obtaining about 2000 images. For the training of the network, we have used the images of 5 and 20x magnification and using a single polarizer.



FIGURE 3 THIN FILM SAMPLES USED



FIGURE 4 ZEISS PETROGRAPHIC MICROSCOPE PRIMOTECH MODEL. GEOLOGICAL TECHNIQUES UNIT. CAI OF EARTH SCIENCES AND ARCHAEOOMETRY. UCM

Once the whole set of images is available, the next step is to find a neural network that, once trained, gives good results for the classification of these images. In this

project several neural networks have been trained, finally choosing two of them, with which a better result has been obtained for this work.

In order to train the network, it is provided with a dataset, which is the set of all the images obtained and which has to be wide enough, so that when it observes an unknown image, it has enough information to be able to classify it within one of the established classes.

In this case, it will be an image of a rock, and it will classify it in one of the quarries that have been provided, indicating the probability that it is from each of them.

Objectives

The main objective of this work is as mentioned, the creation of a rock classification system using neural networks. In turn, this is materialized in a secondary objective of this work which is: the development of a computer application that allows the user to interact with it and obtain classification of rocks from a microscopic image in a simple and friendly way.

To be useful, this application must allow a user to upload an image and obtain the result of the classification of this image in an acceptable time and with the highest possible reliability. The program will have several models loaded, depending on the magnification of the image and the polarizer used. In addition, in the development of the application it is proposed that, within this application, there will be two versions, one in which it is not necessary for the user to register and another one in which he/she will be able to do it and, therefore, will have more functionalities available.

In the version without registration, the user can only choose one of the trained networks, although he will be able to choose different magnifications. In this version, only one image will be allowed to be uploaded, which will be classified and the probabilities that it belongs to the different quarries will be shown in the application. The objective is that the user can test the functionality and usefulness of the system without having to go through a registration process.

However, in the version with registration, the user will be able to choose between the two trained networks, and also different models, depending on the magnifications and the polarizer used. Another advantage of being registered is that several photos can be uploaded simultaneously, which the program will classify at the same time, obtaining from each one of them, simultaneously, the probabilities of belonging to the different quarries. This program will also allow you to download a document with all the results.

As we have said, the main objective is to make it an intuitive and simple program, where the user only has to upload an image, obtaining easily the results of its

classification. Finally, the application should be extendable for future research, i.e. it should be open to the incorporation of new quarries and the consequent (re)training of the networks.

Quarries used

The samples used in this project have been extracted from different historical quarries collected in **Table 1**, it shows the petrographic classification and location of these quarries. Of the quarries chosen, seven of them are of sedimentary rocks of the sandstone type (subarcosa and arcosa) and the eighth of a metamorphic rock, quartzite. Sandstones are detrital sedimentary rocks, which contain at least 50% of sand-sized grains and are composed of quartz and feldspars, as fundamental minerals. Within the sandstones there are several types, of which, two of them have been identified in these quarries, and they are arkosic and subarkosic, which differ in the proportion of the fundamental minerals. Quartzites are metamorphic rocks formed mainly by quartz.

Canteas	Localización	Tipo
Cantera 1	El Polvorín (Toledo)	Subarcosa
Cantera 2	San Antonio (Toledo)	Subarcosa
Cantera 3	Los Molares (Ciudad Real)	Subarcosa
Cantera 4	Pozo paturro (Ciudad Real)	Arcosa
Cantera 5	Arroyo Santo (Jaén)	Subarcosa
Cantera 6	Los Huertos (Soria)	Arcosa
Cantera 7	El Soto (Soria)	Arcosa
Cantera 8	Elounda (Creta, Grecia)	Cuarcita

TABLE 1 QUARRIES ANALYSED AND THE TYPES OF ROCK THEY CONTAIN

Memory organization

The rest of the report is organized as follows:

- Chapter 2 is a tour through the world of artificial intelligence and the different fields that form them explaining and detailing each one of them.
- Chapter 3 is composed by an explanation of the development process of the neural network model that has been used in the project.
- Chapter 4 explains the framework chosen to provide an interface to the application and thus be able to deploy it on a web page.
- Chapter 5 shows the experimental results that have been obtained with the use of the application.
- Chapter 6 the challenges that we have had to face are exposed, as well as the results obtained once the project was completed and finally the distribution of work among each of the members of the group.
- Chapter 7 there is a summary of the bibliography used throughout the document.

Agradecimientos

Las imágenes utilizadas en este trabajo se han realizado en c la Unidad de Técnicas Geológicas del CAI de Ciencias de la Tierra y Arqueometría de la Universidad Complutense de Madrid (UCM). Para su realización han colaborado dos grupos de investigación de la UCM, el grupo “Técnicas cristalográficas y geológicas. Aplicaciones no-convencionales” de la Facultad de Ciencias Geológicas y el grupo ABSYS de la Facultad de Informática. Los equipos utilizados para el entrenamiento de las redes neuronales han sido financiados por la Comunidad de Madrid y la Unión Europea a través del proyecto Micro-Stress-MAP CM. Las redes neuronales aquí desarrolladas forman la base para la continuación del proyecto *Determinación de tensiones residuales microscópicas empleando métodos de difracción, mapas de EBSD, y algoritmos evolutivos.*

2. Sistemas de Clasificación de Imágenes basados en

Inteligencia Artificial

Como es de todos conocido, el desarrollo de la Inteligencia Artificial (IA) durante la última década ha sido espectacular. En todas las ramas de conocimiento han surgido aplicaciones y problemas en los que se utiliza una u otra forma de IA para abordarlos. De la misma forma, este proyecto está centrado en la IA, en concreto, en los sistemas de clasificación mediante tratamiento de imágenes y redes neuronales. Antes de describir nuestra implementación, vamos a introducir algunos conceptos clave que utilizamos en el proyecto. Se partirá de una introducción a la misma, indicando los diferentes subcampos que existen y focalizando en los que son más importantes para este proyecto. También se indicarán los frameworks y hardware utilizados en este campo.

2.1. Introducción a la IA

Como hemos dicho, este proyecto se encuentra dentro del campo de la informática conocido como Inteligencia Artificial (del inglés, Artificial Intelligence, AI).

Algunas de las definiciones dadas de inteligencia artificial por distintos autores son:

- *‘La IA es el arte de crear máquinas con capacidad de realizar funciones que realizadas por personas requieren de inteligencia.’* (Kurzweil, 1990)
- *‘La IA es el estudio de cómo lograr que las computadoras realicen tareas que, por el momento, los humanos hacen mejor.’* (Rich, Knight, 1991).
- *‘La IA es la rama de la ciencia de la computación que se ocupa de la automatización de la conducta inteligente.’* (Lugar y Stubblefield, 1993).
- *‘La IA es el campo de estudio que se enfoca a la explicación y emulación de la conducta inteligente en función de procesos computacionales.’* (Schalkoff, 1990).

Aunque la definición que más se acerca a la realidad de la inteligencia artificial es la creación de programas o máquinas que, dotándolos de ciertos datos de entrada, sean capaces de resolver problemas, para ello imitan funciones cognitivas asociadas al ser humano.

Allan Turing fue el primer científico en hablar sobre Inteligencia artificial cuando, en 1950, inició su artículo “Computing machinery and intelligence”, formulando la pregunta: “¿las máquinas pueden pensar?”. Mediante el conocido como test de Turing lo que se buscaba era medir la capacidad de una máquina para

hacerse pasar por ser humano. En 1956, fue acuñado el término inteligencia artificial por John MacCarthy, Marvin Minsky y Claude Shannon en la Conferencia de Dartmouth.

La inteligencia artificial se divide en varias ramas, de las cuales destacan los sistemas expertos, la robótica, el procesamiento de lenguaje natural, los sistemas de visión, los sistemas de aprendizaje y las redes neuronales. Este proyecto se centrará en las tres últimas.

Los sistemas de visión tienen como objetivo que un computador sea capaz de entender una escena o las características de una imagen. Por su parte, el aprendizaje automático desarrolla técnicas que permiten a los computadores aprender, generalizando comportamientos a partir de una información suministrada.

2.2. ¿Qué es una red neuronal?

Una red neuronal artificial (conocidas como redes neuronales en el mundo de la IA) es un modelo computacional inspirado en las redes neuronales, es decir están basadas en el funcionamiento de las redes de neuronas de organismos vivos. El objetivo es que una máquina pueda dar respuestas similares a las que es capaz de dar el cerebro. Las redes neuronales están compuestas de neuronas artificiales interconectadas entre sí y se agrupan en diferentes capas. En este proyecto se focalizará en las redes convolucionales, que son un tipo de red neuronal con una capa oculta llamada “capas convolucionales”.

Estas capas están formadas por nodos o unidades que se unen mediante conexiones, algunas son denominadas como de entrada, y otras de salida, según reciban los estímulos del exterior o sean ellas las que generen una respuesta. Cada unidad tiene asignado un peso o número, que es, por así decirlo, la memoria de la que dispone esa unidad. Con el tiempo, ese número suele ir modificándose. También disponen de un valor numérico para cada instante de tiempo, que se llama estado o nivel de activación, cuyo cálculo se divide en dos componentes: función de entrada y función de activación.

A pesar de que las redes neuronales buscan tener una arquitectura similar a lo que sería la del propio cerebro humano, su fundamento básico recae sobre la Estadística, ya que la inferencia que se lleva a cabo no es más que cálculos estadísticos y relaciones entre distintos valores que previamente han sido calculados utilizando medidas estadísticas, como puede ser la media. Funciona bien utilizando redes neuronales que ya han sido entrenadas, ya que son capaces de detectar patrones (características) en las imágenes, con un alto grado de precisión, ya que se trabaja a nivel de píxel en una imagen, es un campo que se ha visto ayudado con la capacidad que durante los últimos años

hemos ido ganado para gestionar y tratar grandes cantidades de datos ya que al tener un mayor número de datos los cálculos estadísticos son más precisos

2.3. Principales tipos de redes neuronales

Se va a analizar y clasificar los distintos tipos de redes teniendo en cuenta distintas de las características que los hacen diferentes entre sí.

2.3.1. Clasificación de redes neuronales según la topología de la red.

La clasificación de las redes se puede realizar observando las características de las principales capas que la forman, a continuación, se analizarán varias:

- Red neuronal *Monocapa*, es la red neuronal más simple, compuesta por una capa de neuronas de entrada, que se conecta a una capa de neuronas de salida, en la cual, se realizan los cálculos.
- Red neuronal *Multicapa*, es una generalización de la red neuronal monocapa, lo que diferencia a esta red es que dispone de un conjunto de capas intermedias (capas ocultas), entre la capa de entrada y la de salida. Estas capas pueden estar total o parcialmente conectadas.
- Red neuronal *convolucional* (CNN), la principal diferencia está en que cada neurona no se une con todas las capas siguientes, sino que se une sólo con un subgrupo de ellas (se especializa).
- Red neuronal *recurrente* (RNN), estas redes no tienen una estructura de capas, sino que permiten conexiones arbitrarias entre las neuronas.

2.3.2. Clasificación de redes según el método de aprendizaje.

A continuación, se analizan los distintos tipos de redes según su método de aprendizaje (**Figura 5**):

- El aprendizaje supervisado es una técnica que utiliza datos de entrada y de salida y, en función de estos datos, realiza predicciones. Es decir, intenta encontrar una función para clasificar cada entrada en alguna de las etiquetas de salida.
 - Aprendizaje por corrección de error, cambia los pesos y analiza la diferencia entre los valores esperados y los

obtenidos, para así, ajustar dichos pesos, en función de ese error cometido.

- Aprendizaje estocástico, realiza cambios aleatorios sobre los pesos, y se queda con aquellos cambios que mejoren la predicción.
- El aprendizaje no supervisado no dispone de datos de salida para poder clasificar la entrada. Intenta encontrar similitudes entre los datos de entrada.

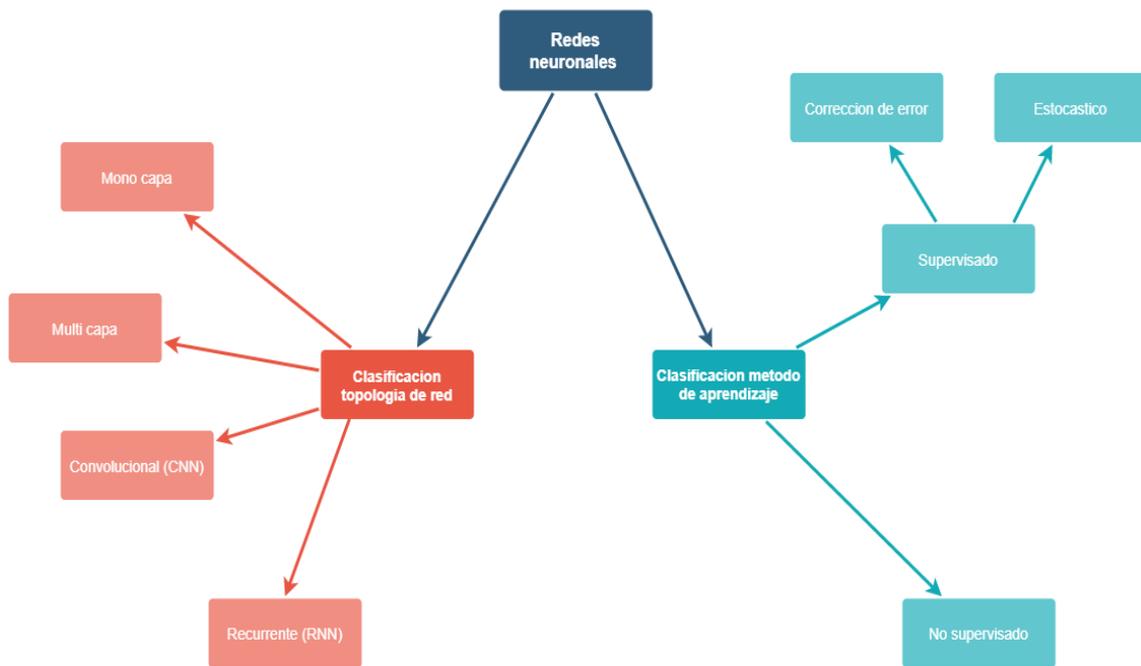


FIGURA 5 TIPOS Y SUBTIPOS DE REDES NEURONALES

2.4. Redes convolucionales

Las redes convolucionales se utilizan principalmente para analizar imágenes.

Lo que diferencia a una red convolucional de otras redes neuronales, son las capas ocultas llamadas “capas convolucionales”. Con estas capas puede detectar patrones en imágenes. Estos patrones se determinan mediante el uso de filtros en la operación convolucional.

En las redes convolucionales existen dos componentes principales, las capas convolucionales y las capas de agrupación.

El funcionamiento de una red convolucional se lleva a cabo tomando una imagen con un tamaño y un número de canales de color. Se aplica un filtro determinado, llamado kernel, que tendrá un tamaño. El filtro convolucionará con la imagen de entrada y generará una imagen destacada.

La idea principal de una capa convolucional es algo parecido a filtros, es decir, vectores que pueden ser de múltiples dimensiones que se aplican a la imagen de entrada una vez ha sido convertida esta misma en vectores también así se puede llevar a cabo la operación de convolución.

2.5. Red AlexNet

En este proyecto se ha utilizado la red convolucional AlexNet (**Figura 6**).

Este modelo de red AlexNet hace uso de una arquitectura CNN que consta de cinco capas y tres capas “fully connected”, con sesenta millones de parámetros y seiscientos cincuenta mil neuronas.

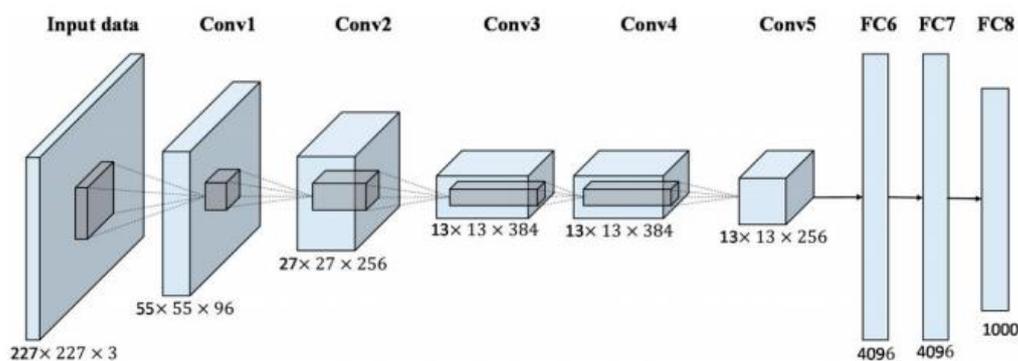


FIGURA 6 ARQUITECTURA DE LAS CAPAS QUE FORMAN LA RED ALEXNET

La primera capa de convolución está formada por 96 filtros, las siguientes dos capas están a su vez seguidas por una capa “max-pooling”, la tercera, cuarta y quinta capa están conectadas directamente y después sigue una capa “max-pooling”, otra vez. A continuación, están las tres capas “fully connected”, y en esta última se aplica la función softmax, para poder hacer la predicción en un conjunto de datos multiclase.

Después de cada capa convolucional y cada capa “fully-connected”, se aplica la función ReLu como función de activación. Para evitar un ajuste excesivo, se utiliza la técnica del aumento de datos y la técnica dropout. [2.]

2.5.1. Capas convolucionales

Para realizar el “procesamiento distintivo” (**Figura 7**) de las redes neuronales, se harán llamadas convolucionales, que consisten en tomar “grupos de píxeles cercanos” de la imagen de entrada, y operar con una matriz llamada kernel. Se asume que ese núcleo tiene 3x3 píxeles, logrando con este tamaño, “visualizar” todas las neuronas de entrada y así, generar una matriz de salida, que será la nueva capa de neuronas ocultas.

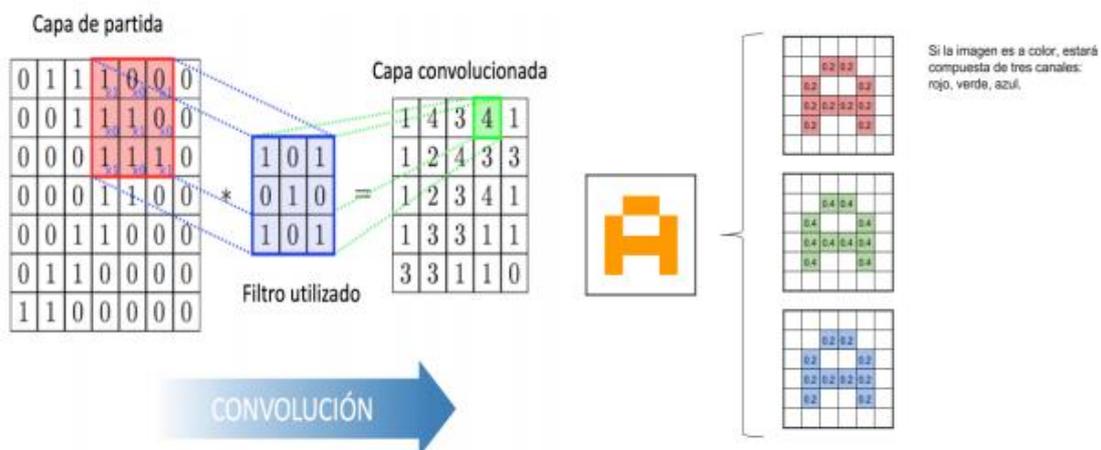


FIGURA 7 PROCESO DE CONVOLUCIÓN

2.5.2. Capas “fully-connected”

Al final de una red neuronal convolucional, hay una o más capas conectadas, cuando están “completamente conectadas”, cada uno de los nodos en la primera capa está conectado a cada uno de los nodos de la segunda capa. (**Figura 8**)

La clasificación se realiza basándose en los atributos extraídos en las convoluciones.

Normalmente, la última capa “fully-connected” contiene una función softmax de activación, que produce un valor de probabilidad que va desde 0 hasta 1, para cada una de las etiquetas de clasificación sobre las cuales el modelo va a inferir.

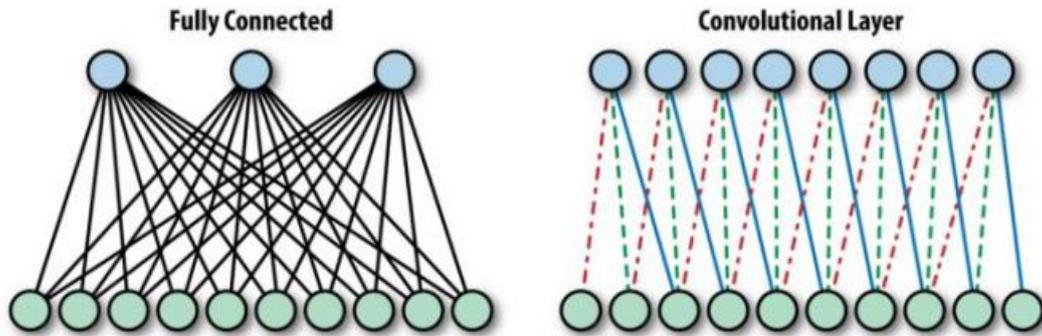


FIGURA 8 DIFERENCIAS ENTRE CAPAS FULLY CONNECTED Y CONVOLUCIONALES

2.5.3. Función ReLu

La función ReLu (**Ecuación 1**) transforma los valores introducidos, anulando los valores negativos y dejando los valores positivos, a medida que se ingresan.

$$f(x) = \max(0, x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$$

ECUACIÓN 1 RELU

2.5.4. Función Softmax

La función Softmax se implementa a través de una red neuronal, justo antes del resultado. La capa Softmax (**Figura 9**) debe tener el mismo número de nodos que el resultado capa.

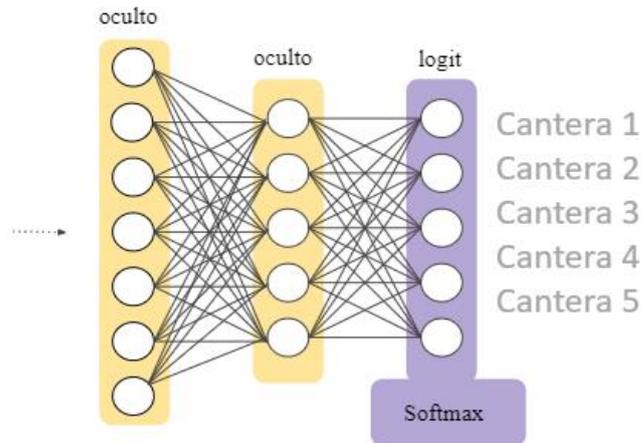


FIGURA 9 CAPA SOFTMAX

2.5.5. Data Augmentation

El aumento de datos es la generación artificial de datos mediante alteraciones en los datos originales. Esto permite aumentar el conjunto de datos de entrenamiento, tanto en tamaño como en diversidad. **(Figura 10)**

En visión artificial, esta técnica se convirtió en un estándar de regularización, para mejorar el rendimiento y combatir el sobreajuste en las CNN.

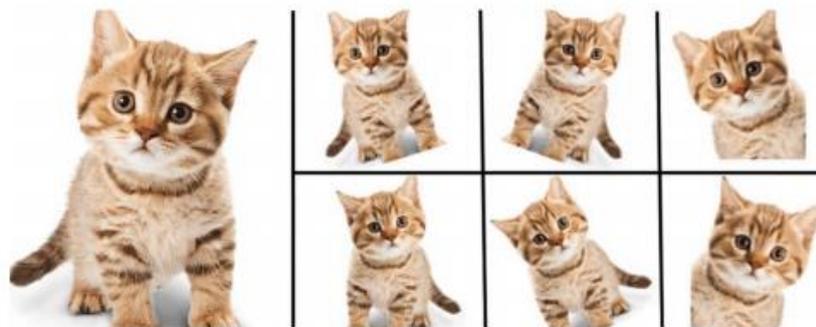


FIGURA 10 EJEMPLO DATA AUGMENTATION

2.5.6. Pooling

Cuando la capa convolucional genera una nueva imagen más pequeña, se puede aplicar a esa nueva imagen una agrupación **(Figura 11)**, que consta de otro filtro que reducirá la imagen, agrupando varios píxeles en uno. Esto ayuda a no realizar un sobreajuste y reducir la carga de cálculo requerida en la siguiente capa.

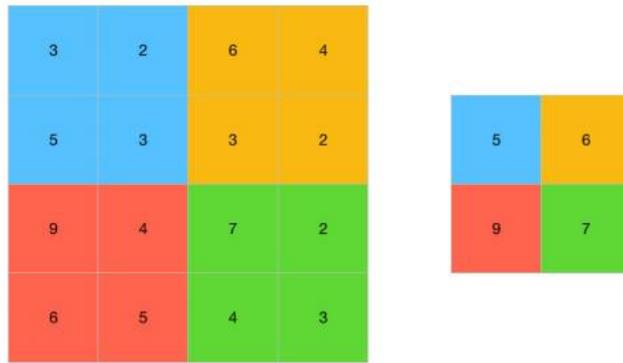


FIGURA 11 EJEMPLO POOLING

2.5.7. Técnica DropOut

Para cada nueva entrada a la red en el entrenamiento fase, un porcentaje de las neuronas en cada capa oculta, será aleatoriamente desactivado (**Figura 12**), de acuerdo con una probabilidad de caída, previamente definida.

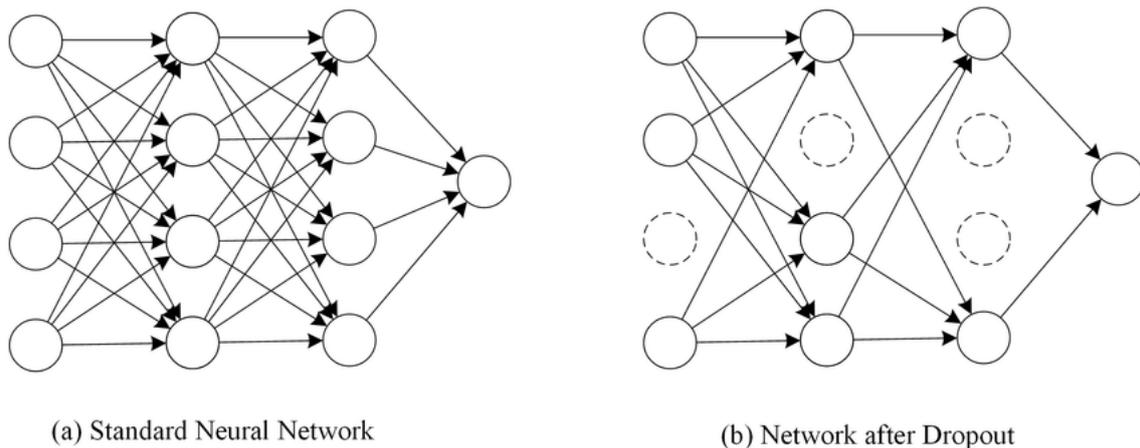


FIGURA 12 EJEMPLO DROPOUT

2.6. Deep Learning

Deep Learning es un subconjunto de Machine Learning que, a su vez, es un subcampo de la informática donde las máquinas aprenden a realizar tareas por sí mismas, sin haber sido antes programadas explícitamente para ello. Las máquinas observan un patrón e intentan imitarlo, razonando y sacando sus propias conclusiones. Es la forma de aprendizaje más similar a la del ser humano, ya que se basa en el funcionamiento del cerebro y las conexiones entre neuronas. Por eso, Deep Learning utiliza redes neuronales, que se componen de diferentes niveles jerárquicos, donde cada nivel está conectado para ir transmitiendo información de uno a otro. El primer nivel toma una información sencilla, que transmite al siguiente nivel, este otro nivel la combinará para

transformarla en otra más compleja y enviarla al siguiente nivel y así sucesivamente hasta llegar al último, donde se obtendrá la salida. [3.]

2.7. Detección, clasificación y segmentación

Dentro del campo de la visión artificial se emplean diferentes técnicas según la imagen que se quiere analizar. Estas técnicas se emplean en el conjunto de datos utilizados, llamado dataset.

Algunas de las técnicas más frecuentemente utilizadas son:

- La *clasificación* de imágenes es una técnica que se aplica al conjunto de imágenes de entrada. Lo que pretende es clasificar las imágenes en diferentes categorías o clases. Cuando se aplica esta técnica, se obtiene una salida, la cual indica, para cada imagen, qué porcentaje de cada clase se encuentra en ésta.
- La *localización* señala en la imagen dónde se encuentran los objetos de las clases etiquetadas, es decir, delimita los objetos de las diferentes clases, dentro de la imagen.
- La *detección* de objetos busca unificar las dos técnicas anteriores, para precisar la información acerca de la imagen.

2.8. Framework Nvidia Digits y Pytorch.

Un framework es un sistema de software que sirve de base para el desarrollo del mismo. Éste ayuda a tener una mejor organización y hace más sencillo la escritura y el desarrollo de aplicaciones, facilitando la organización, el desarrollo y la reutilización en un futuro.

Existen varios tipos que se pueden adaptar a las necesidades del proyecto que se quiere realizar.

Uno de los tipos que existen son los frameworks Open Source, que son software de código abierto, es decir, el código fuente se publica de forma abierta, bajo una licencia de uso libre, que permite emplearlo y modificarlo.

Entre los más utilizados se pueden destacar:[4.]

- Tensorflow: es un framework de código abierto para el aprendizaje automático. Cuenta con herramientas, bibliotecas y recursos que permiten innovar y aplicar con facilidad procesos de Inteligencia Artificial.
- Caffe: es un marco de aprendizaje profundo creado pensando en la expresión, la velocidad y modularidad.
- Torch: es un marco informático científico y un lenguaje de script basado en el lenguaje de programación Lua. Proporciona una gran variedad de algoritmos para el aprendizaje profundo.

En este proyecto se ha utilizado la biblioteca PyTorch, que está basada en la biblioteca Torch, utilizada principalmente para aplicaciones como la visión por computador y el procesamiento del lenguaje natural y desarrollada principalmente por el laboratorio de inteligencia artificial de Facebook. Es un software gratuito y de código abierto publicado bajo la licencia BSD modificada. [5.]

PyTorch proporciona dos funciones de alto nivel:

- Computación tensorial (como NumPy) con fuerte aceleración a través de GPU.
- Redes neuronales profundas construidas sobre un sistema de diferenciación automática basada en tipos.

PyTorch cuenta con una interfaz sencilla para la creación de redes neuronales pese a trabajar de forma directa con tensores, sin la necesidad de una librería a un nivel superior.

Uno de los aspectos más importantes en las tareas de aprendizaje automático es el hardware empleado. Ya que el conjunto de datos puede ser de gran tamaño, los procesos podrían tardar un tiempo elevado.

Nvidia Digits (**Figura 13**) es un sistema de formación de GPU de aprendizaje profundo. Se utiliza para entrenar una red neuronal profunda de alta precisión para tareas de clasificación de imágenes, segmentación y detección de objetos. [6.]

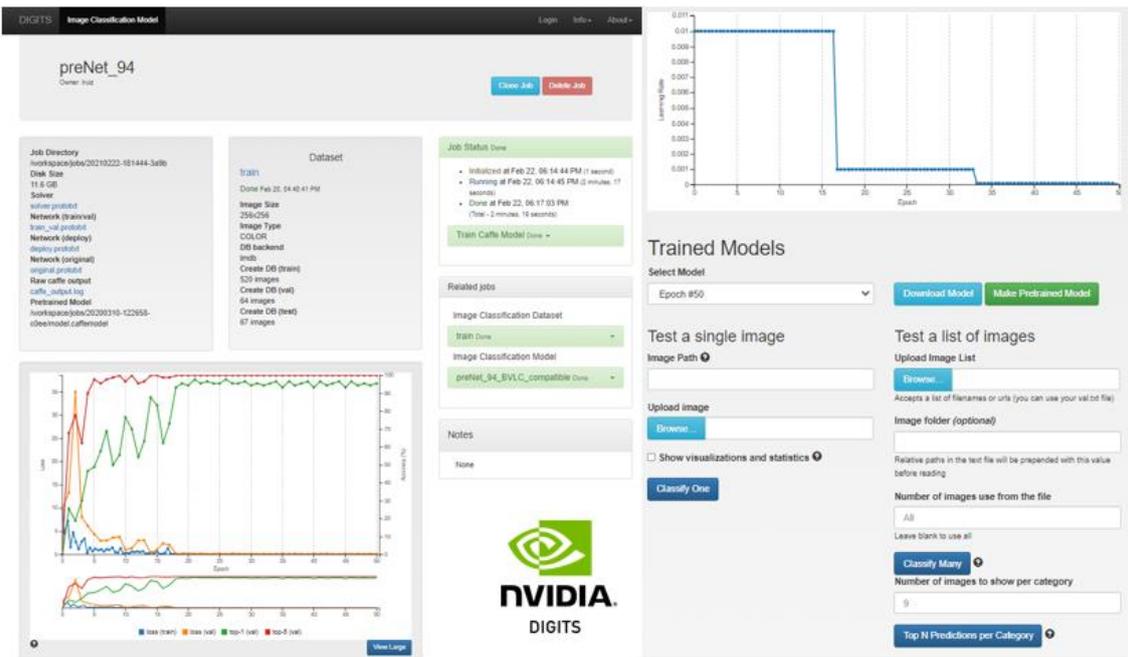


FIGURA 13 GUI NVIDIA DIGITS EJEMPLO DE UN MODELO ENTRENADO

3. Metodología de trabajo y desarrollo del proyecto

El corazón del proyecto es una red neuronal entrenada para realizar las predicciones y esta NN se puede consultar a través de una aplicación web que hemos denominado StoNEt, nombre que aplicamos igualmente a la NN. A continuación, explicaremos el desarrollo de la NN y posteriormente, (cap 4) explicaremos las características de la aplicación web.

3.1. Desarrollo del Modelo de red neuronal.

Lo primero que se ha de tener en cuenta a la hora de comenzar el desarrollo de un modelo de red neuronal, son las características que debería tener el sistema y la complejidad del problema que se va a intentar resolver con él.

3.2. Diseño e investigación

Para realizar este proyecto hemos llevado a cabo una investigación previa de cuáles eran las principales redes neuronales más utilizadas, así como el rendimiento que tenía cada una de ellas y los usos que solían tener. Una vez realizada esta búsqueda, analizando la red que más se adecuaba a nuestras necesidades y que tenía un equilibrio entre coste computacional / precisión, nos decantamos por la red AlexNet, que en anteriores apartados ha sido explicada más en detalle. Tras elegir la red, comprobamos cómo de bien lo hacía con un dataset de ejemplo extraído de internet y observamos si había alguna característica que pudiéramos modificar para mejorar el rendimiento de la red, observamos que no eran necesario aplicar ninguna técnica para incrementar el rendimiento como podría ser data augmentation así que finalmente utilizamos la estructura de la red original únicamente reentrenándola para nuestro problema en cuestión.

3.3. Obtención del conjunto de datos

Una vez tenemos definida la estructura de la red, el siguiente paso a realizar consiste en la obtención del conjunto de datos que, posteriormente, sería utilizado como dataset para entrenar la red y así conseguir unos resultados adecuados para nuestra casuística. Para ello tuvimos que realizar numerosas sesiones en la Unidad de Técnicas Geológicas, primero para realizar un entrenamiento del funcionamiento del microscopio petrográfico a cargo del personal técnico del centro donde fuimos realizando más de 2000 imágenes de las láminas delgadas de las muestras de las ocho canteras históricas que nos servirán de datos.

De todas las imágenes obtenidas se seleccionaron las imágenes obtenidas con cinco y veinte aumentos, respectivamente y un solo polarizador, buscando obtener la mayor variabilidad entre muestras, ya que no conocíamos con cual le iba a resultar más fácil al modelo aprender las características de las distintas clases sobre las que debería realizar la inferencia.

En la **Figura 14** se pueden observar dos imágenes de las muestras obtenidas para la cantera 1, a la izquierda, tomada con 5 aumentos y a la derecha, utilizando 20 aumentos. En la esquina inferior derecha las imágenes están incluida la escala con la que puede deducirse el número de aumentos que se han utilizado a la hora de tomar la imagen. Las imágenes obtenidas tienen formato .tiff y un tamaño de 12 mb.

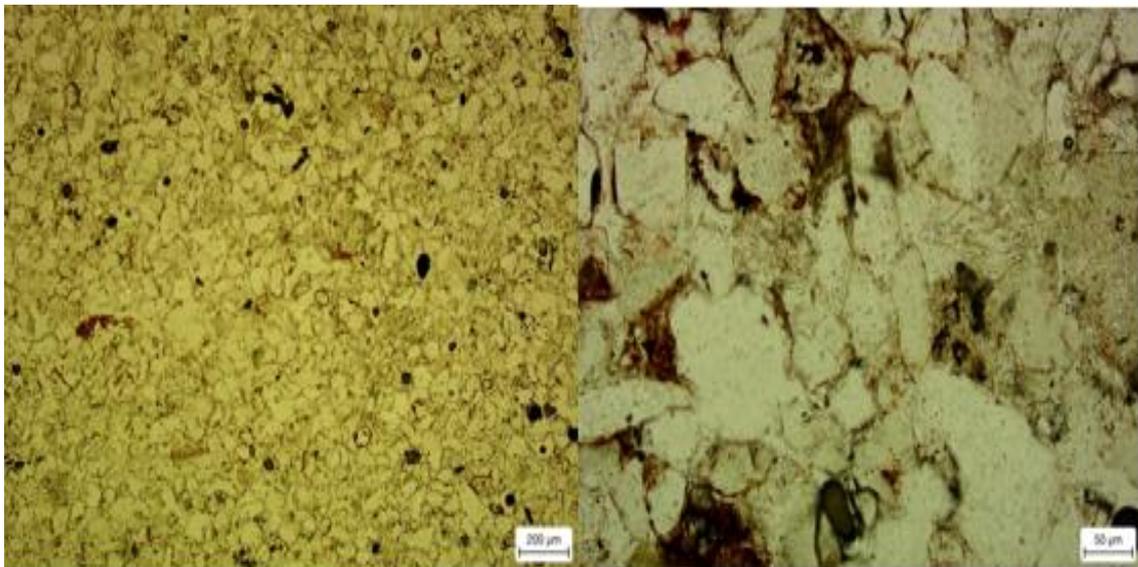


FIGURA 14 MICROGRAFÍAS DE LA CANTERA 1 (EL POLVORÍN, TOLEDO) OBTENIDAS CON LOS OBJETIVOS DE 5X Y 20X.

3.3. Normalización y clasificación

Tras haber realizado el duro “trabajo de campo”, la siguiente etapa en el desarrollo fue llevar a cabo la clasificación de las distintas muestras, tomadas tanto por clases como por características de la imagen (aumentos y polarizadores), así conseguimos el denominado dataset, que es finalmente el conjunto de datos con los cuales el modelo aprende las distintas características y peculiaridades a nivel de píxel de cada clase. También tuvimos que normalizar el formato de las imágenes, ya que el microscopio nos devolvía un archivo con extensión .tiff (utilizada para el archivo en bruto sin perder información pero que ocupa mucho espacio), pero queríamos que nuestro modelo trabajase con

imágenes en jpg, puesto que ocupan menos y nos iba a ser menos costoso computacionalmente el entrenamiento con ellas.

Para la creación de los dataset hemos utilizado en un primer paso y utilizando la plataforma de Nvidia Digits, el propio sistema que ella te ofrece, donde automáticamente e indicando el porcentaje de imágenes que quieres utilizar para entrenamiento, testing y validación, se te genera una base de datos con ellas. Con Pytorch tuvimos que hacerlo en varias líneas de código donde, una vez creados los directorios de carpetas, se le pasaba el path directamente y así se hacía un preprocesado de las imágenes previo a generar los dataloader. También sería en esta parte del proceso donde se llevaría a cabo el denominado data augmentation, aunque, por las características estudiadas, era de mayor interés que la imagen no hubiese sufrido ninguna modificación a nivel de forma o ángulo.

3.5. Implementación de la transferencia de aprendizaje

Una vez que se ha generado correctamente el dataset y los dataloader, se debe cargar el modelo pre entrenado que se va a utilizar, en este caso, la red AlexNet. Con ello, lo que se busca es no tener que inicializar los parámetros previos a un valor para el cual se define un rango, sino utilizar los valores de una red que, previamente, ha sido entrenada con millones de imágenes, lo que consigue que sea capaz de distinguir una mayor variedad de características en las imágenes. De esta forma, se estaría transfiriendo el conocimiento adquirido en una tarea más genérica, a este caso particular, por lo que no se comienza de cero. Tras esto, los pesos del modelo deben congelarse, ya que no interesa que sean modificados. También se deberán modificar las capas del modelo que interesen, en este caso, la última, ya que es la responsable de dar el resultado, para así adaptarlo a nuestro número de clases a evaluar (**Figura 15**).

Una vez que se ha generado el dataset correctamente y los dataloader debemos cargar el modelo pre entrenado que vamos a utilizar en nuestro caso la red AlexNet ,con ello lo que buscamos es que todos los parámetros previos que anteriormente debían ser inicializados a un valor para el cual se define un rango , ahora utilizamos los valores de una red que previamente ha sido entrenada con millones de imágenes lo que consigue que sea capaz de distinguir una mayor variedad de características de las propias imágenes , de esta forma estaríamos transfiriendo el conocimiento adquirido en una tarea más genérica a nuestro caso particular por lo que no comenzamos de cero. Tras esto y como nos interesa que los pesos del modelo no sean modificados debemos congelarlos, también deberemos cambiar las capas que nos interesen del modelo en este caso la

última para adaptarlo a nuestro número de clases a evaluar ya que es esta capa la responsable de dar el resultado.



FIGURA 15 TRANSFER LEARNING ALEXNET – STONET

3.6. Entrenamiento

Cuando se tiene el modelo definido, lo siguiente es cargarlo en el dispositivo donde se va a llevar a cabo el entrenamiento, pudiendo ser en la GPU o en la CPU. También es muy importante definir tanto el criterio, como el optimizador que serán utilizados. En este caso, se ha utilizado como criterio, la función de pérdida de entropía cruzada y como optimizador, el de Adam, que suele ser el más utilizado para los problemas de clasificación multiclase.

3.7. Métricas de evaluación y análisis de los resultados

Las principales métricas de clasificación usadas en el Machine Learning [7.] son:

- Matriz de confusión.
- Precision o accuracy.
- Sensibilidad o recall.
- Pérdida logarítmica.

Accuracy y Loss

Son conceptos que se utilizan para el entrenamiento de redes neuronales, con el objetivo de controlar el incremento de mejora que sufre una red por cada etapa del entrenamiento. Si la arquitectura utilizada es la correcta, incluso se puede detectar casos en los que se está produciendo un sobreajuste, lo que puede ocasionar pérdidas en el rendimiento del modelo.

Son conceptos que se utilizan durante el entrenamiento de redes neuronales para poder controlar el incremento de mejora que sufre una red por cada etapa del entrenamiento, si la arquitectura utilizada es la correcta incluso poder detectar casos en los que se está produciendo un sobreajuste (**Figura 16**) lo que puede ocasionar pérdidas en el rendimiento de nuestro modelo.

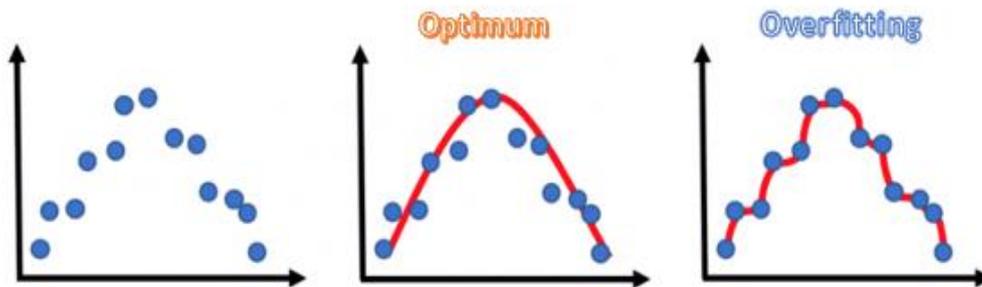


FIGURA 16 EJEMPLO DE OVERFITTING

En esta imagen se pueden observar tres gráficas, que muestran los resultados de un entrenamiento. La gráfica de la derecha es un ejemplo de modelo sobreentrenado, ya que se genera una curva con más ruido y más irregular. Lo ideal es obtener una curva lo más parecida posible a la figura del medio.

En esta imagen podemos observar tres gráficas como resultado de un entrenamiento, un ejemplo de modelo sobreentrenado sería la gráfica de la derecha ya que se genera una curva con más ruido y más irregular, lo que nos interesa es obtener una curva lo más parecida posible a la figura de en medio.

Matriz de confusión

En este proyecto se ha utilizado como medida la matriz de confusión, que es una tabla que describe el rendimiento de un modelo supervisado de Machine Learning en los datos de prueba, donde se desconocen los verdaderos valores. Se llama “matriz de confusión” porque hace que sea fácil detectar dónde el sistema está confundiendo varias clases. Existen otros indicadores para evaluar el rendimiento:

- True Positives (TP): cuando la clase real del punto de datos era 1 (Verdadero) y la predicha es también 1 (Verdadero)
- False Positives (FP): cuando la clase real del punto de datos era 0 (False) y el pronosticado es 1 (True).
- False Negatives (FN): Cuando la clase real del punto de datos era 1 (Verdadero) y el valor predicho es 0 (Falso).

El escenario ideal es que el modelo dé 0 False Positives y 0 False Negatives. La **Figura 17** ilustra lo que sería una matriz de confusión.

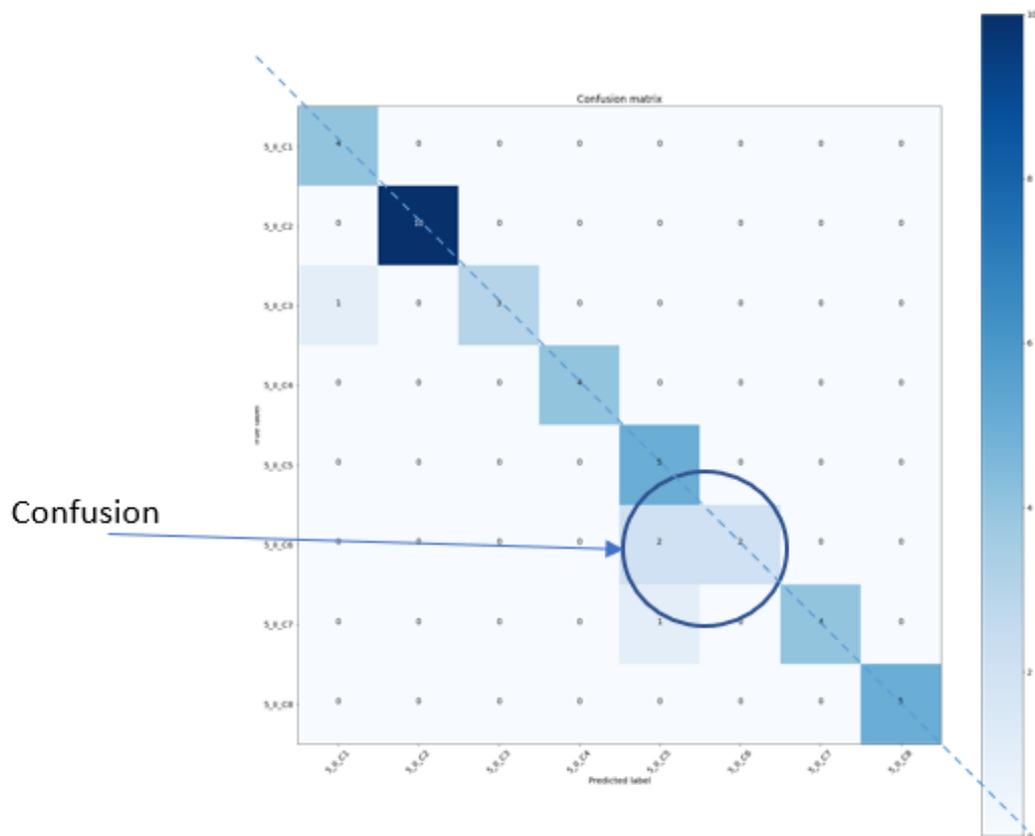


FIGURA 17 EJEMPLO DE MATRIZ DE CONFUSIÓN

Este ejemplo ha sido obtenido en una de las primeras etapas del modelo que está siendo utilizado en la aplicación del proyecto. Como puede observarse, en la gran mayoría de los casos, están alineados los resultados con las clases correctas (diagonal de la matriz), pero también puede observarse como las clases 5 y 6 pueden ser confundidas en alguna ocasión. Quitando esas clases, se puede ver que los resultados están bastante alineados con los resultados esperados para cada uno.

3.8. Generación del modelo NN final

Tras conseguir un rendimiento adecuado para el propósito que se quería conseguir, es necesario guardar el modelo, y así poder utilizarlo con las imágenes de muestras que se quieran clasificar. Aquí se muestra una gráfica que contiene los principales pasos (**Figura 18**) a seguir para el desarrollo de un modelo de red neuronal.

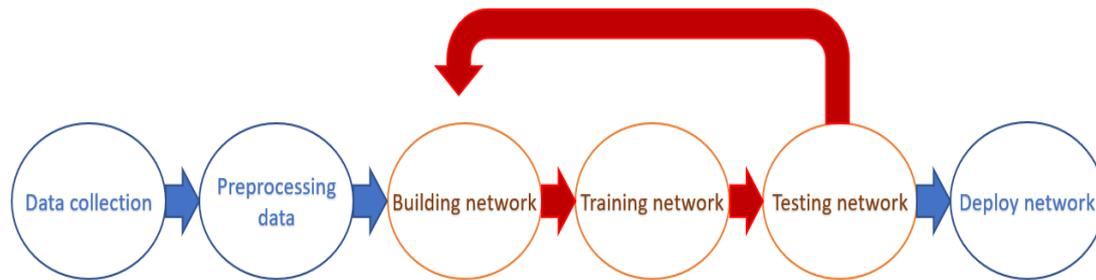


FIGURA 18 ETAPAS EN EL DESARROLLO DE UN MODELO DE RED NEURONAL

3.9. Rendimiento del entrenamiento en distintos dispositivos y con diferentes frameworks

El rendimiento es un aspecto que, en cualquier campo de la Informática, puede suponer un factor diferencial entre que algo funcione correctamente o no (timings), en concreto para la Inteligencia Artificial, el rendimiento es el responsable de que una red pueda ser ejecutada en un dispositivo, como puede ser un SOC, o sea necesario de una computadora más potente, con el coste económico que eso supondría. Por ello se ha decidido llevar a cabo los entrenamientos en dos dispositivos distintos, comenzando en una GPU, que es donde comúnmente se suelen llevar a cabo los entrenamientos hoy en día, ya que sus capacidades para este trabajo son mayores (memoria, cálculo de matrices y paralelismo) y, para terminar, con la propia CPU.

Las pruebas (**Figura 19**) han sido realizadas tanto en una gpu Nvidia RTX 2060, como en una cpu AMD Ryzen 7 4800H con 8 cores y 16 threads. Estas pruebas han sido realizadas utilizando un mismo batch para ambos dispositivos, siendo cierto que, por las características de los dispositivos, el tamaño del mismo ha sido limitado por la CPU, cuya capacidad era menor que la de la GPU, puesto que esta última dispone de más VRAM, permitiendo generar batch de tamaños mayores. El dataset utilizado también ha sido el mismo.

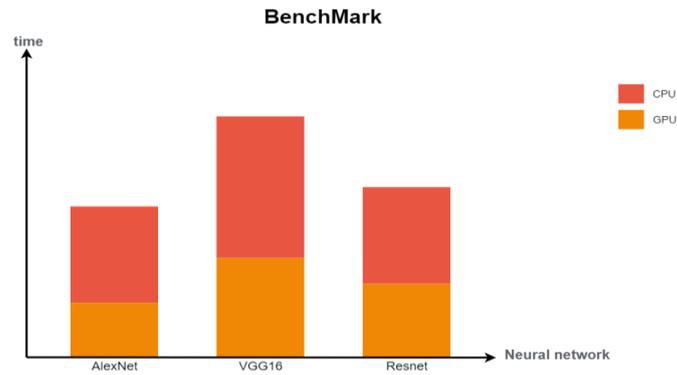


FIGURA 19 BENCHMARK NETWORKS DEVICES

Como puede observarse en la **Figura 20**, el framework utilizado a la hora de llevar a cabo el desarrollo de la red neuronal, también juega un papel muy importante, puesto que algunos están mejor optimizados que otros, haciendo que los tiempos de ejecución sean mucho menores. La gráfica muestra las diferencias entre los principales frameworks existentes hoy en día.

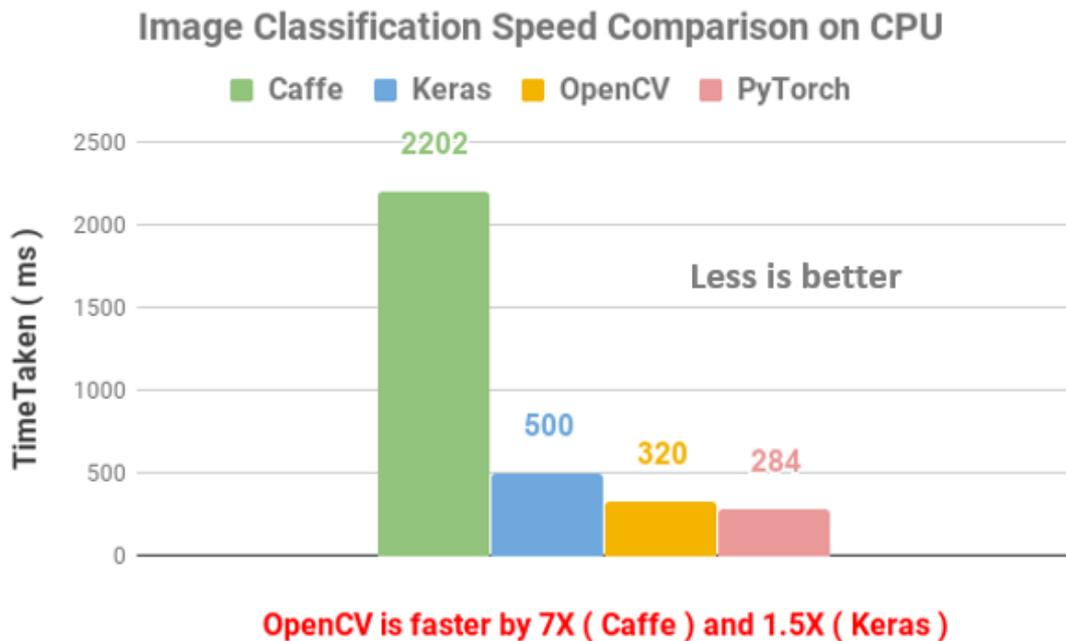


FIGURA 20 BENCHMARK FRAMEWORKS

4. Desarrollo aplicación web GUI con Flask: 'StoNet'

Hasta ahora, cuando se quería obtener la clasificación de una muestra dada por el modelo utilizado, se debía acceder mediante terminal y, con un conjunto de comandos, se podía obtener el resultado de la inferencia realizada.

Pero, llegados a este punto, lo que se pretende es que cualquier persona con conocimientos mínimos en informática, sea capaz de sacar provecho de este desarrollo y así, hacer que este proyecto pueda llegar a más público. Para ello, se plantea la creación de una página web, donde el usuario pueda subir sus datos y obtener los resultados de manera sencilla. Como la mayor parte del proyecto ha sido desarrollada en Python, se ha buscado un framework que facilitase la labor a la hora de poner en funcionamiento una web, con un back hecho en Python, así se llegó a Flask [8.], que es un framework minimalista escrito en Python, que permite crear aplicaciones web rápidamente y con un mínimo número de líneas de código. Está basado en la especificación WSGI de Werkzeug y el motor de templates Jinja2 y tiene una licencia BSD. Tras familiarizarse con la documentación y buscar ejemplos, se consiguió poner en marcha la web y, poco a poco, ir añadiendo funcionalidades (**Figura 21**).

La aplicación web se divide principalmente en dos bloques, uno que se dedica a toda la gestión del login y usuarios, y otro que es la parte encargada de pasar y formatear los datos correctamente, para que estos lleguen al modelo y así, este sea capaz de devolver un resultado que, nuevamente, tendrá que tratarse, para así poder mostrarlo al usuario por pantalla.

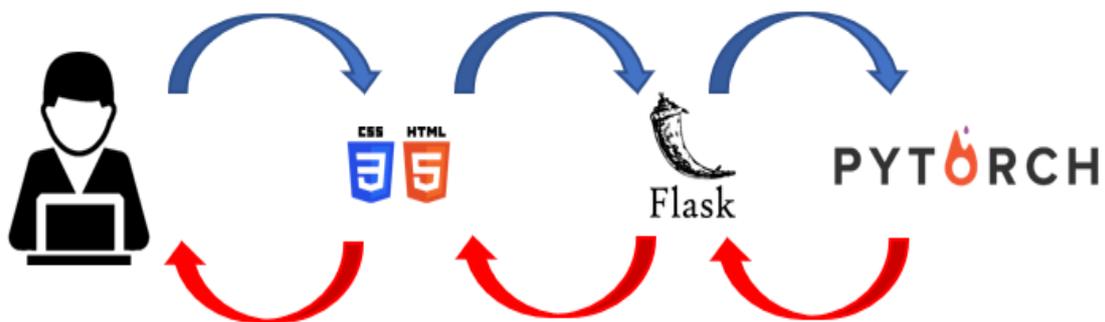


FIGURA 21 CAPAS DE LA APLICACIÓN

4.1. Despliegue del modelo

En la **Figura 22** se muestra un pequeño flujo de la información entre la aplicación web y el propio modelo, pudiéndose observar cuales son los principales archivos responsables en el proceso de inferencia.

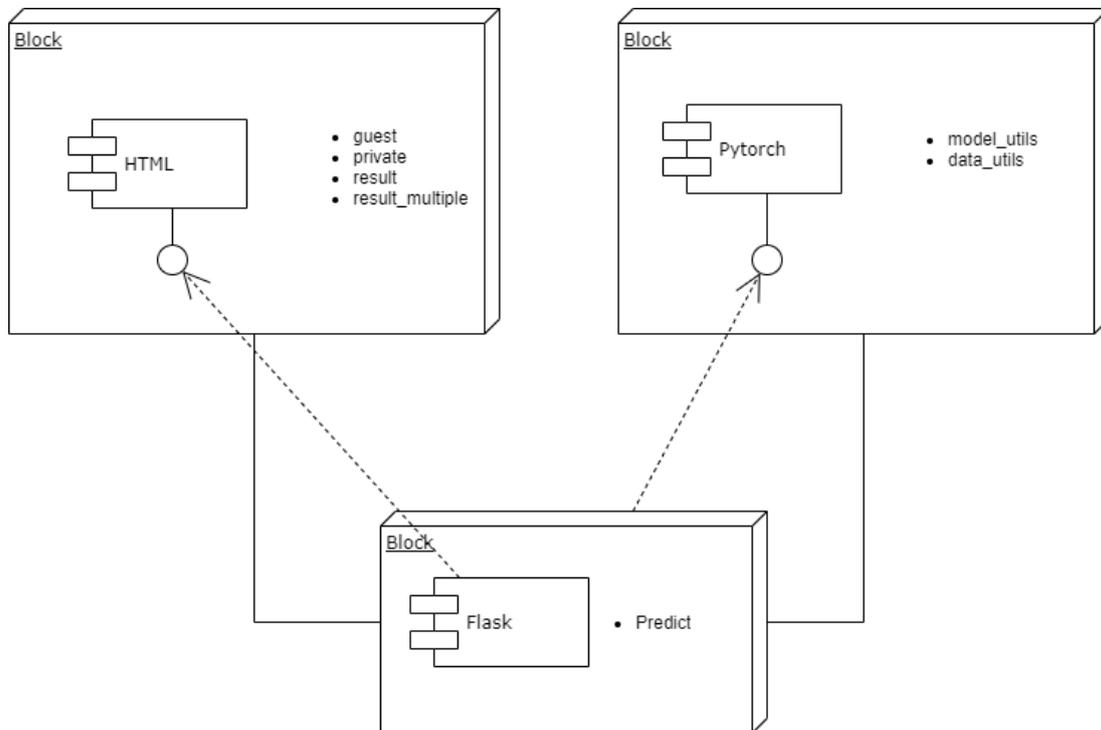


FIGURA 22 MÓDULOS DE LOS QUE FORMAN LA APLICACIÓN

Sin duda alguna, la parte más complicada de este proceso ha sido la de conseguir comunicación entre la parte front de la aplicación, como es el html [9.] ,css [10.] y flask Python [1.] por debajo, con la parte del back, donde se hace uso de Pytorch y otras librerías necesarias en el proceso.

4.2. Depuración del uso y funcionamiento de la aplicación.

Una vez se tuvo integrado el modelo en la aplicación web, el siguiente paso a seguir fue el de testear y depurar los posibles errores propios del incorrecto tratamiento de la información, así como la realización de posibles acciones por parte del usuario de manera malintencionada, con el único fin de conseguir que la aplicación produzca algún error.

Se han utilizado métodos de seguridad con ayuda del paquete Werkzeug de python, como la generación de códigos hash para no almacenar las contraseñas en texto plano, y para todo el sistema de login, se ha hecho uso del paquete por defecto que ofrece Flask, llamado flask login.

También se ha intentado encapsular todo lo referente al login mediante el uso de Blueprints, pudiendo así decidir qué partes de la aplicación eran accesibles únicamente para el usuario que había hecho login y cuales eran accesibles para todos los usuarios.

4.3 Despliegue sobre un servidor.

Para poder finalizar con el TFG y hacer accesible nuestra aplicación hemos tenido que desplegarla en una máquina de la UCM, este proceso se ha realizado con ayuda para poder tener los permisos necesarios de ejecución, tanto del servidor de la aplicación como de la base de datos. A fin de hacer más fácil el proceso de puesta en marcha de la aplicación en cualquier servidor se ha creado un fichero de requisitos llamado *requirements.txt*, donde se encuentran todos los paquetes de python necesarios y las versiones que han sido utilizadas en el proyecto, con esto se consigue que de forma automática y una vez teniendo python 3 instalado ejecutamos la orden `pip3 install -r requirements.txt` se instale todo.

La url a través de la cual se puede acceder a esta aplicación es la siguiente:

stonet.dacya.ucm.es

5. Resultados experimentales.

5.1 Entrenamiento de la red

Para llevar a cabo el entrenamiento de las distintas redes en el proyecto, se han utilizado imágenes que previamente habían sido tomadas por nosotros. Exactamente han sido 842 para la red que trabaja con 20 aumentos - filtro paralelo y 577 para la red que trabaja con 5 aumentos - filtro paralelo. En el total de imágenes están incluidas tanto las de que han sido tomadas para el entrenamiento, así como para la verificación y también para el testing. Se han utilizado 50 etapas en cada entrenamiento.

Entrando en detalle en cada una de ellas, comenzaremos con la gráfica que corresponde al modelo que ha sido entrenado utilizando una red AlexNet y con imágenes de 5 aumentos (**Figura 23**). En la gráfica puede observarse que en el eje X (abcisas) se representa el número total de etapas por las que ha transcurrido el entrenamiento, y en el eje Y (ordenadas) tenemos tanto los valores de Loss y Accuracy. En la gráfica hay 4 curvas que hacen referencia, por un lado de color rojo y amarillo al accuracy tanto de Training como de Validation obtenido en cada etapa del entrenamiento y por otro lado de color azul y verde tenemos las correspondientes a los valores de Training Loss y Validation Loss .

Como se puede observar las curvas correspondientes al accuracy y al loss rápidamente divergen lo que indica que la red comienza a aprender desde las primeras etapas, el hecho de que las curvas del accuracy no se separen indican que estamos utilizando una arquitectura de red adecuada al problema que estamos intentando solucionar. Realmente y como se puede observar por los datos de las gráficas los valores **máximos de accuracy y los mínimos de loss** se dan a mitad del entrenamiento por lo que no sería necesario que fuese tan largo ya que a partir de esas etapas la mejora que se produce es ínfima. Puede existir cierto **sobreajuste (Overfitting)**, que queda reflejado cuando las curvas de validation y training de un mismo parámetro (loss o accuracy) divergen y se separan, pero realmente no hemos considerado que sea demasiado.

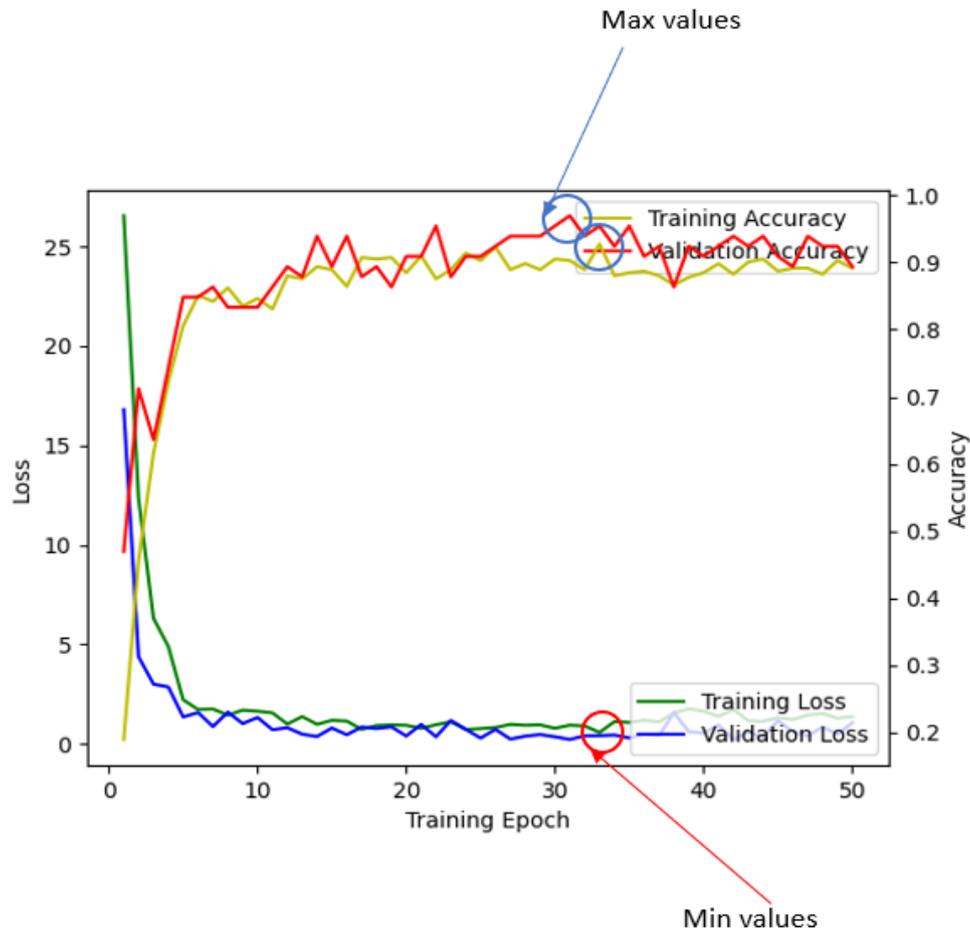


FIGURA 23 GRÁFICA ENTRENAMIENTO ALEXNET X5 AUMENTOS

A continuación, tenemos la gráfica para el entrenamiento de una red utilizando AlexNet e imágenes de 20 aumentos (**Figura 24**), los datos que se muestran en ella son los mismos que podemos observar en la anterior figura. En este caso sí que se puede detectar un mayor nivel de sobreajuste ya que las curvas se distancian más entre ellas. Llegados a este punto decidimos que sería más útil utilizar un conjunto de datos realizado con cinco aumentos lo que permitía a la red tener una mayor capacidad de aprendizaje generando un menor sobreajuste en el proceso.

AlexNet 20 aumentos 0.86 accuracv

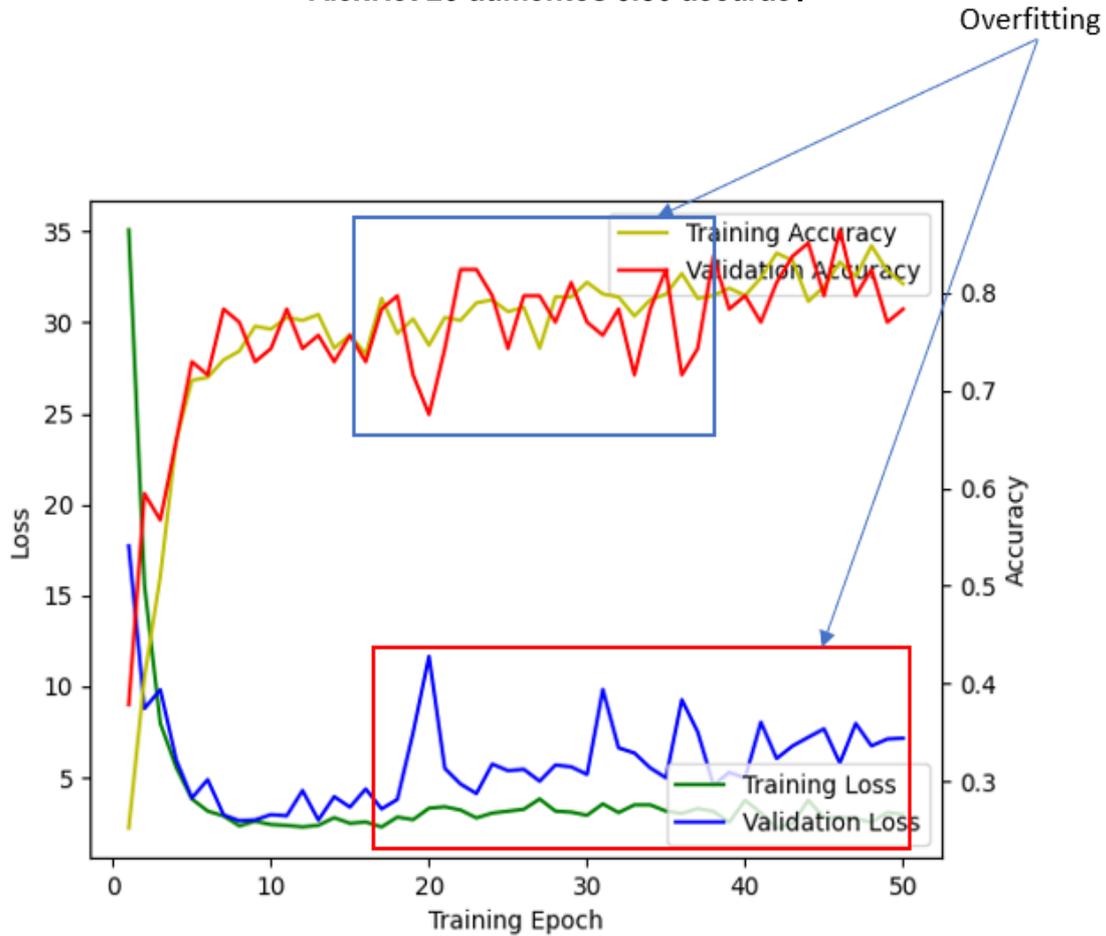
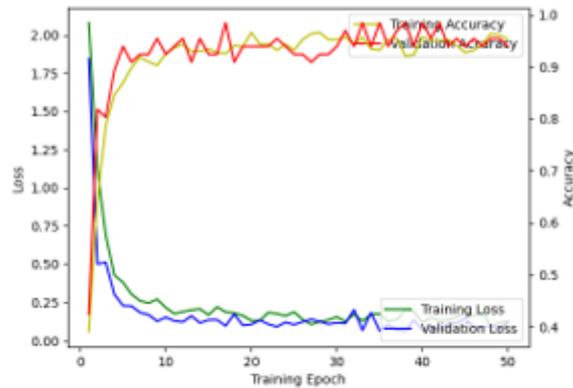


FIGURA 24 GRÁFICA ENTRENAMIENTO ALEXNET X20 AUMENTOS

En la **Figura 25** se recogen las gráficas correspondientes a sendos entrenamientos con las imágenes de cinco y veinte aumentos, utilizando la red ResNet. Esta ofrece un accuracy similar a AlexNet pero genera algo menos de overfitting.

ResNet 5 aumentos 0.96 accuracy



ResNet 20 aumentos 0.89 accuracy

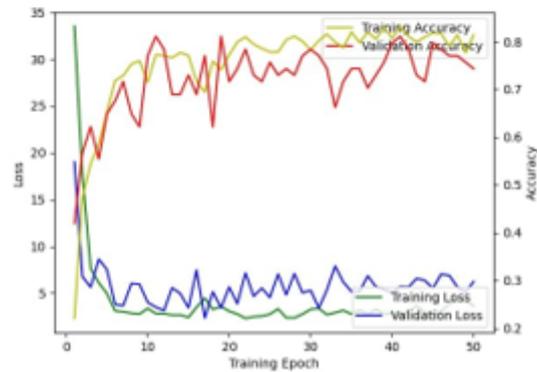


FIGURA 25 GRÁFICAS DE ENTRENAMIENTO DE RESNET X5 Y X20 AUMENTOS

5.2 Prueba de clasificación de imágenes no utilizadas en el entrenamiento

A continuación se realizó una inferencia (**Figura 26**) con una muestra sin clasificar que se nos facilitó de una piedra de afilar y pudimos deducir que la cantera con la que mayor similitud tenía era la 8, aunque no se disponía de suficientes imágenes para sacar una conclusión con un alto grado de confianza.

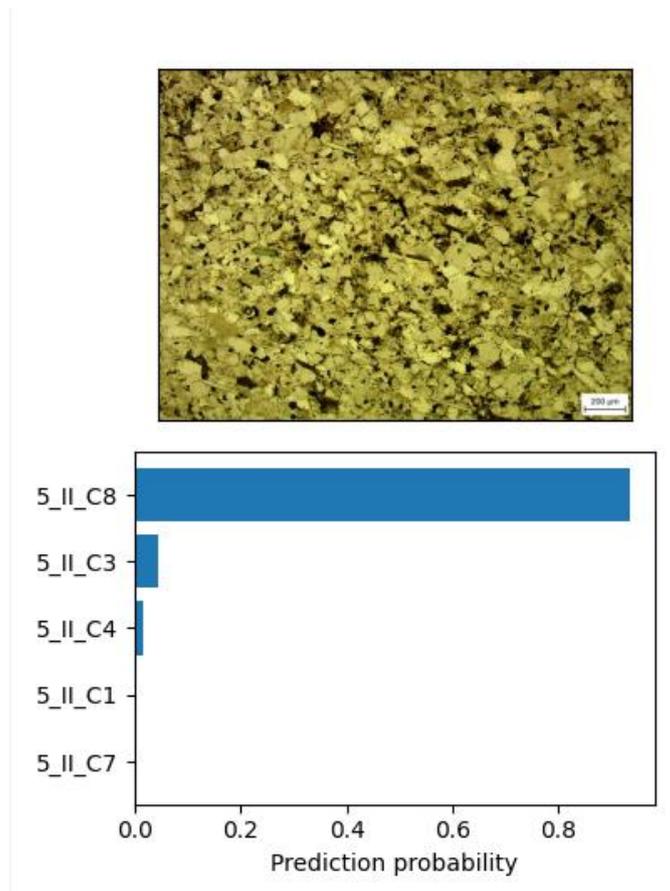


FIGURA 26 INFERENCIA RESULTADO TRAS ANALIZAR UNA MUESTRA

La aplicación también dispone de la posibilidad de realizar una análisis de muestras múltiples, ya que con una sola consulta es capaz de realizar la inferencia sobre varias muestras como se puede observar en la **Figura 27** la precisión es muy elevada en las muestras cargadas de la cantera 2 con las que no había sido entrenada la red previamente obteniéndose un 90% de aciertos, lo que también nos hace ver la necesidad de subir varias muestras para obtener un resultado con mayor confianza. Este resultado a diferencia de la versión con una

sola muestra ofrece la opción de descargar un informe en excel con la probabilidad mayor para cada muestra enviada.

	A	B	C	D	E	F	G
1	Model./project/models/alexnetModel511.pth						
2	Muestra	Cantera	Probabilidad				
3	IMG_1532.jpg	5_II_C2	0.89				
4	IMG_1534.jpg	5_II_C6	0.99				
5	IMG_1536.jpg	5_II_C2	1.0				
6	IMG_1538.jpg	5_II_C2	1.0				
7	IMG_1540.jpg	5_II_C2	1.0				
8	IMG_1542.jpg	5_II_C2	1.0				
9	IMG_1544.jpg	5_II_C2	0.83				
10	IMG_1546.jpg	5_II_C2	1.0				
11	IMG_1548.jpg	5_II_C2	1.0				
12	IMG_1549.jpg	5_II_C2	1.0				
13							
14							
15							
16							
17							

FIGURA 27 EXCEL RESULTADO DEL ANÁLISIS MÚLTIPLE

6. Conclusiones

En este trabajo se ha buscado desarrollar una aplicación que sirviese como herramienta para los colectivos cuyo trabajo forma parte de la clasificación y análisis de las rocas. Con esta finalidad ha sido creada la aplicación StoNet que permite facilitar y agilizar el trabajo de clasificación de muestras mineralógicas. Las principales conclusiones de este trabajo han sido:

- Los sistemas de clasificación basados en Redes Neuronales constituyen una herramienta de gran potencial en el campo de la geología.
- Los sistemas de Transfer Learning permiten reutilizar el entrenamiento previo de redes de gran tamaño y adaptarlo al problema particular de la clasificación de rocas.
- Las redes implementadas, una vez entrenadas permiten la clasificación de nuevas imágenes en tiempos de ejecución mínimos, compatibles con el desarrollo de un sistema de clasificación on-line y off-line.
- Es posible establecer un compromiso entre el coste computacional y el tamaño de la red, dados los sistemas de cómputo disponibles.

Tras concluir con este proyecto hemos constatado las grandes capacidades que tiene la Inteligencia Artificial y Visual, ya que, utilizando una arquitectura adecuada a las necesidades del problema, un conjunto de datos lo suficientemente amplio y equilibrado, y una interfaz con la que poder interactuar con ella es posible conseguir resultados muy satisfactorios para clasificar imágenes de forma automática.

6.1. Retos a los que nos hemos tenido que enfrentar.

- Investigar para conseguir averiguar qué campo de la inteligencia artificial era el adecuado para resolver el problema al que nos enfrentamos.
- Entender el machine learning para poder desarrollar un modelo con la suficiente precisión para que sea realmente fiable el resultado que nos proporciona.
- Desplegar este modelo de NN en una aplicación web y al mismo tiempo ser “user friendly” para cualquier nivel de conocimiento informático.
- Familiarizarnos con el lenguaje de programación Python ya que era totalmente desconocido para nosotros.

- Conseguir un buen resultado equilibrado entre coste computacional y complejidad (tipo de red pre entrenada AlexNet vs Vgg16), aquí podemos ver una (**Tabla 2**) con una las distintas redes neuronales existentes, así como su precisión y el coste computacional expresado en FLOPs (operaciones en punto flotante)

Comparison					
Network	Year	Salient Feature	top5 accuracy	Parameters	FLOP
AlexNet	2012	Deeper	84.70%	62M	1.5B
VGGNet	2014	Fixed-size kernels	92.30%	138M	19.6B
Inception	2014	Wider - Parallel kernels	93.30%	6.4M	2B
ResNet-152	2015	Shortcut connections	95.51%	60.3M	11B

TABLA 2 TABLA COMPARACIÓN DE RENDIMIENTO ENTRE DISTINTAS REDES NEURONALES

- Organizar y distribuir el trabajo ha sido otro gran reto, ya que junto a nuestros tutores hemos tenido que marcar timings y entregas.

6.2. Aportación de cada uno de los miembros

Luis Ruiz de Nicolas

Ha participado en todas las partes del proyecto.

- Para comenzar, como se ha expuesto anteriormente, hubo un trabajo previo, que consistió en la toma de imágenes de muestras en el laboratorio del CAI. Para ello, asistió varios días al laboratorio, donde tuvo que familiarizarse con el microscopio petrográfico y la aplicación que se usaba para la toma de imágenes a través de este. En cada sesión realizó una serie de imágenes de cada muestra, utilizando láminas delgadas proporcionadas por el laboratorio, realizando un barrido completo de cada lámina. Estas imágenes se guardaban en una aplicación, que también proporcionó el laboratorio, donde se guardaban directamente del microscopio.
- A continuación, realizó una investigación sobre el tema que se iba a tratar, leyendo libros, artículos y manuales para la realización del proyecto. En conjunto con los directores de este proyecto, se fueron realizando tareas, avanzando poco a poco en este trabajo de investigación. Se informó sobre los estudios que se estaban haciendo de la parte de geología, donde aprendió sobre las diferentes canteras utilizadas, las propiedades de cada, cómo realizaban el trabajo antes de empezar este proyecto y qué querían obtener con él. También investigó sobre el campo de la inteligencia artificial y la rama en la que se iba a centrar este proyecto, realizando con los directores unas tareas semanales que se iban mostrando y corrigiendo.
- Después de tener esta información, empezó con el desarrollo del modelo de red neuronal. Hizo pruebas con distintas redes, probando primero con redes que no estaban entrenadas, empezando a entrenarlas desde cero, lo que no dio muy buenos resultados. Así, siguió probando con redes preentrenadas, obteniendo mejor precisión, por lo que decidió utilizar estas últimas. Las redes se entrenaron con imágenes de todas las canteras, utilizando 20 y 5 aumentos y filtro paralelo. Después de probar con varias de estas redes, eligió la red AlexNet, ya que era con la que mejores resultados obtuvo. Una vez elegida la red, el siguiente paso fue estudiar su arquitectura, para poder ajustar detalles con los que obtener

mejores resultados al clasificar las imágenes. El entrenamiento de estas redes lo realizó en un principio utilizando Caffe en Nvidia Digits y por último Python, utilizando para ello PyTorch concretamente.

- Tras tener un modelo plenamente funcional en la labor de clasificación de las imágenes de muestra, el siguiente paso fue el desarrollo de una parte capaz de comunicarse con la interfaz de la aplicación, para lo que se utilizó Flask, que ayuda en el proceso de integración de Python en una aplicación Web, ya que todas las peticiones generadas y los formularios necesitaban un punto de acceso para poder interactuar con el modelo de red neuronal.
- Participó en el desarrollo de la interfaz.

Sara Núñez Sánchez

Ha participado en todas las partes del proyecto.

- Primero, realizó un trabajo previo, que trataba de recopilar imágenes de todas las muestras de las canteras que se iban a utilizar en el proyecto. Para ello, asistió varios días al laboratorio del CAI de Geología, donde tuvo que familiarizarse con el microscopio petrográfico, con el que se realizó la toma de muestras, y con la aplicación que se utilizaba para guardar estas imágenes.
Se tomaron imágenes de láminas delgadas de todas las canteras, proporcionadas por el laboratorio, haciendo un barrido de estas, realizando una serie de imágenes con 5, 10 y 20 aumentos, y filtros paralelos y cruzados.
- El siguiente paso fue investigar sobre el proyecto que se iba a realizar, recopilando información sobre los temas que se iban a tratar. En conjunto con los directores del proyecto, se realizaron tareas semanales, en las que se iba recogiendo información sobre los diferentes campos que se iban a tratar. Por una parte, se investigó sobre la parte de geología, donde se informó sobre el trabajo que estaban realizando y qué querían conseguir con este proyecto. Se investigó sobre las canteras que se iban a utilizar, por qué materiales están formadas, cuáles son las técnicas de extracción, etc., cómo realizan los estudios tradicionalmente y qué se podía hacer para mejorar eso con este proyecto. Por otro lado, investigó sobre la parte de inteligencia artificial y los campos de esta en los que se iba a centrar este proyecto, qué métodos se iban a utilizar, qué redes existen y cuáles eran más eficientes para lo que se quería conseguir.
- Contribuyó en el entrenamiento de las redes.
- Se ha encargado del desarrollo de la parte visual de la aplicación. En una primera versión, se utilizó una interfaz con un diseño propio que, posteriormente, tuvo que ser modificada para seguir un estilo alineado al resto de proyectos con los que, en un futuro, será integrada. Esta parte se ha desarrollado utilizando HTML y CSS, ya que las peticiones se hacen mediante las estructuras de formularios que, por defecto, ofrece HTML.
- Se ha buscado un diseño responsive en toda la aplicación, ya que es muy importante que, si el usuario hace uso de ella en un dispositivo

portátil, como puede ser una Tablet o un móvil, la experiencia en la navegación no se vea afectada de forma negativa. Ya que en el laboratorio recogen las imágenes de las muestras realizadas con el microscopio a través de una aplicación instalada en una Tablet, se ha pensado en un diseño que sea accesible a través de esta, y así, directamente poder subir las imágenes realizadas a nuestra aplicación desde este dispositivo.

6. Conclusions

In this work we have sought to develop an application that would serve as a tool for groups whose work is part of the classification and analysis of rocks. For this purpose, the StoNet application has been created to facilitate and speed up the work of classification of mineralogical samples.

6.1. Challenges we have had to face

- Research to find out which field of artificial intelligence was the right one to solve the problem we were facing.
- To understand machine learning in order to develop a model with sufficient accuracy to provide reliable results.
- Deploy this NN model in a web application and at the same time be "user friendly" for any level of computer literacy.
- Familiarize ourselves with the Python programming language as it was totally unknown to us.
- To achieve a good balance between computational cost and complexity (type of pre-trained network AlexNet vs Vgg16), here we can see with a **(Table 2)** the different existing neural networks, as well as their accuracy and computational cost expressed in FLOPs (floating point operations).

Comparison					
Network	Year	Salient Feature	top5 accuracy	Parameters	FLOP
AlexNet	2012	Deeper	84.70%	62M	1.5B
VGGNet	2014	Fixed-size kernels	92.30%	138M	19.6B
Inception	2014	Wider - Parallel kernels	93.30%	6.4M	2B
ResNet-152	2015	Shortcut connections	95.51%	60.3M	11B

TABLE 2 PERFORMANCE COMPARISON TABLE BETWEEN DIFFERENT NEURAL NETWORKS

- Organizing and distributing the work has been another great challenge, since together with our tutors we have had to set deadlines and deliveries.

7. Bibliografía

- [1.] Estudio López-Andrés et al. 2022 (en prensa)
- [2.] “Hacia una caracterización de las piedras de afilar naturales y sus canteras fuente históricas en la Península Ibérica” por S. López-Andrés, V. López-Acevedo Cornejo, X. Arroyo Rey y Pedro P.R. Moya-Maleno en el libro “Al otro lado del filo: armas, herramientas y piedras de afilar desde la prehistoria”. “Hacia una caracterización de las piedras de afilar naturales y sus canteras fuente históricas en la Península Ibérica” en (en prensa))
- [3.] Trask, Andrew W. (2019). Grokking Deep Learning. Manning Publications Co.
- [4.] Frameworks <https://www.threepoints.com/es/Frameworks-de-IA-que-debes-conocer>
- [5.] <https://medium.com/pytorch>
- [6.] <https://docs.nvidia.com/deeplearning/digits/digits-user-guide/index.html>
- [7.] <https://towardsdatascience.com/20-popular-machine-learning-metrics-part-1-classification-regression-evaluation-metrics-1ca3e282a2ce>
- [8.] <https://flask.palletsprojects.com/en/2.0.x/#user-s-guide>
- [9.] <https://www.w3schools.com/html/>
- [10.] <https://www.w3schools.com/css/>
- [11.] <https://flask.palletsprojects.com/en/2.0.x/>
- [12.] <http://adilmoujahid.com/posts/2016/06/introduction-deep-learning-python-caffe/>
- [13.] <https://machinelearningmastery.com/how-to-develop-a-convolutional-neural-network-to-classify-photos-of-dogs-and-cats/>
- [14.] <https://www.diegocalvo.es/clasificacion-de-redes-neuronales-artificiales/>
- [15.] <https://cleverpy.com/que-es-pytorch-y-como-se-instala/>
- [16.] Ketkar, Nikhil y Moolayil, Jojo (2021). Deep learning with Python: learn best practices of deep learning models with PyTorch. Apress.
- [17.] Goodfellow, Ian, Bengio, Yoshua y Courville, Aaron (2016). Deep Learning. MIT Press.
- [18.] Berzal, Fernando (2019). Redes Neuronales & Deep Learning - Volumen 1: Entrenamiento de redes neuronales artificiales. Independently published.