RECONOCIMIENTO DE PIEZAS DE AJEDREZ

Redes Neuronales
Máquinas de soporte vectorial
Árboles de decisión
K-vecinos

Javier Fernández Rozas
Marcos Díaz Estévez
Tomás Villalba Ferreiro
Xoel González Pereira
Noelia Suárez Tuñas

Grupo de prácticas: 3.2

Aprendizaje Automático
Universidade da Coruña
Curso 2022/23

Índice

1. Introducción

El ajedrez ha sido, y es, uno de los juegos más analizados y estudiados en el campo de la Inteligencia Artificial. Con un número de posiciones posibles siendo 10^{120} , y contando con variables como el pensamiento abstracto, táctico o razonamiento lógico, el ajedrez se postula como un candidato perfecto para probar diferentes enfoques de la IA (Heurísticas de evaluación, Redes Neuronales...)

Entre los trabajos más destacados, destacan DeepBlue, la primer máquina de ajedrez capaz de ganar al ser humano, Stockfish, el módulo más fuerte, creado usando Heurísticas hechas a mano, y AlphaZero, el que fue capaz de derrotar a Stockfish usando Aprendizaje por refuerzo.

Continuando con estas ideas, proponemos un Sistema Basado en Conocimiento (SBC) que sea capaz de reconocer las distintas piezas en un tablero de ajedrez.La idea para desarrollar este sistema surge de la intención de facilitar el aprendizaje y la búsqueda de información sobre ajedrez a aquellas personas más inexpertas y a las que ya cuentan con algún conocimiento sobre el tema, respectivamente.

Para ello, se proponen situaciones como las de detectar posiciones de las piezas en tableros en libros sobre el tema, escanear jugadas, permitir que personas ajenas a las partidas las puedan seguir en tiempo real, proporcionar un aprendizaje más cómodo a niños y/o aquellas personas que no tienen un conocimiento muy amplio en el tema, ... Para resolver el problema, utilizaremos una Red de Neuronas Artificiales (RR.NN.AA.), Máquinas de Soporte Vectorial (SVM) o inglés Support Vector Machine, k-vecinos próximos (kNN), en inglés k-Nearest Neighbors. En las RR.NN.AA., probaremos con distintas arquitecturas, de forma que nos quedaremos con la configuración que mejores resultados obtenga y mejor se adapte al problema.

2. Descripción del problema

2.1. Descripción concreta del problema a resolver

El problema a resolver consiste en la implementación de un detector de piezas de ajedrez, mediante el uso del lenguaje Julia. A partir de una base de datos proporcionada, formada por distintas piezas de ajedrez extraídas de diferentes tableros, el sistema entrenado detectará cada una de las piezas en un tablero.

Esta identificación se realizará a partir de fotografías realizadas con el móvil, en la que es posible que se detecten diferencias en las posiciones de las imágenes, su nitidez y otros aspectos. Además, el sistema reconocerá cada tipo de pieza y su color.

Para decidir los mejores hiper parámetros y los mejores modelos usaremos la métrica de precisión.

2.2. Restricciones que se aplican al problema a resolver

Para la correcta identificación de las piezas, se recomienda que las imágenes tomadas del tablero sean lo más rectas posibles, teniendo en cuenta que el tablero tiene que aparecer completo y sin estar recortado, además de una buena calidad de imagen y de luz. Con relación a los escaques (casillas del tablero), solo se aceptarán casillas blancas o las que tengan rayas diagonales negras.

2.3. Descripción de la base de datos

2.3.1. Origen

La base de datos proporcionada está formada por una serie de tableros que han sido extraídos de diferentes libros de partidas magistrales de ajedrez. Cada tablero ha sido fotografiado por cada uno de los integrantes del grupo, sujetándose a las restricciones descritas en el apartado anterior, aunque puede haber ligeras diferencias de imagen debido a que estas se han realizado con diferentes dispositivos.

La distancia a la que han sido tomadas las imágenes es aproximadamente de 15 centímetros. La mayoría de las imágenes suelen tener una resolución de entre 80 y 90 píxeles en alto y 90 píxeles en ancho y las piezas pueden ser de color blanco o negro. Cada pieza puede tener un total de dos fondos diferentes (fondo con diagonales negras o blanco). Para el análisis de las imágenes, estas se han convertido a una escala de grises.

2.4. Propiedades de los datos

2.4.1. Estadísticas

Como ya se mencionó anteriormente, la base de datos estará conformada por imágenes que varían en calidad y en la posición en la que se tomaron, dando lugar a un amplio abanico de tipos datos. Esto se debe sobre todo a que estas fotografías fueron realizadas por distintas personas con distintos dispositivos, principalmente móviles y cámaras profesionales. Debido a que las imágenes han sido tomadas de diferentes libros, el contorno de algunas piezas es ligeramente distinto. Por lo demás, nuestra BD no presentará otras propiedades con cambios, pues el color será siempre en blanco y negro y el tamaño de las imágenes será siempre el mismo (teniendo en cuenta las pequeñas variaciones que pueden surgir al tomar las fotografías desde diferentes ángulos), esto se puede apreciar

en la Figura 1. Peones de la BD. Por ahora no se ha considerado ningún atributo acotado.

2.4.2. Propiedades especiales

Todos las propiedades de los datos se mencionaron en los apartados anteriores, más en concreto en el apartado de estadísticas (2.4.1).

En caso de que la foto en cuestión no cumpla con los estándares de calidad dados previamente, esa foto será tachada como inválida, y no usada en el entrenamiento.

Figura 1: Peones de la BD

3. Bibliografía / Estado del arte

3.1. Trabajos recientes e importantes en el ámbito tratado

Se han realizado algunos proyectos similares al tratado, entre los que se puede resaltar [1] , por A. Laskowski, utilizando Julia y la biblioteca de aprendizaje automático Flux. El propósito de este proyecto es el reconocimiento automático de piezas de ajedrez a través de imágenes de alta calidad. El proyecto se basa en una red neuronal convolucional que ha sido entrenada con miles de imágenes de tableros de ajedrez en diferentes posiciones y perspectivas. La red neuronal es capaz de identificar y localizar cada una de las piezas de ajedrez en la imagen, y luego clasificarlas en función de su tipo (por ejemplo, rey, dama, torre, etc.).Utiliza técnicas avanzadas de procesamiento de imágenes y aprendizaje profundo para lograr una alta precisión en la detección de las piezas. Además, se han implementado varias técnicas de optimización para mejorar el rendimiento de la red neuronal, como la normalización de lotes y el aumento de datos.

El proyecto mencionado anteriormente es un gran ejemplo y fuente de información para este proyecto, pero no se han encontrado muchos más trabajos de gente reconocida o que fueran publicados. Sin embargo, se han encontrado trabajos de investigadores o gente no destacada en el tema, como puede ser [2], por A. A. del Barrio García, para la Facultad de Informática de la Universidad

Complutense de Madrid. En este trabajo, el autor aborda el reto tecnológico de la digitalización automática de partidas de ajedrez mediante visión artificial. El objetivo es facilitar la retransmisión en línea y el análisis de partidas mediante motores de ajedrez. Para ello, se han utilizado varias redes neuronales convolucionales para la clasificación de piezas de ajedrez y se ha acelerado la detección del tablero y la inferencia de estos modelos con una Nvidia Jetson Nano. El resultado es un framework funcional que digitaliza automáticamente la configuración de un tablero de ajedrez en menos de 5 segundos, con una precisión del 92 % al clasificar las piezas y un 95 % al detectar el tablero. Con esto se pretende reflejar la amplia variedad de los documentos e información en cuanto a al conocimiento y experiencia de los autores.

En esta misma línea encontramos el trabajo de fin de grado de Molina Nuñez, A. , Plaza García-Abadillo, C. , Rengel Lazcano, K. para la facultad de Informática de la Universidad Complutense de Madrid [4]. El reconocimiento de las piezas de ajedrez en un tablero físico es un problema de visión que aún no se ha resuelto de manera eficiente. Se han logrado algunos avances a lo largo de los años, pero aún carecen de la precisión o el rendimiento suficientes para usarse a nivel práctico para digitalizar juegos en vivo. Este proceso de digitalización actualmente se realiza a nivel profesional con la ayuda de tableros de ajedrez y piezas especializadas que no están al alcance de la mayoría de los aficionados al ajedrez debido a su costo. Hay un interés renovado en encontrar nuevos algoritmos para este problema, gracias al éxito de las redes neuronales convolucionales. Sin embargo, aún existen varias dificultades para su adopción práctica, especialmente en implementaciones que intentan satisfacer las restricciones de tiempo real de los juegos en vivo con recursos de hardware de presupuesto limitado. En este TFG, han tratado de superar algunas de estas limitaciones portando a una aplicación de Android algunos de los algoritmos más prometedores propuestos en este campo. Además, también han intentado mejorar el entrenamiento de las redes neuronales diseñando e implementando un dispositivo de bajo coste que simplifica la generación del dataset de entrenamiento.

En esta misma línea también encontramos el proyecto de investigación [3], por M. Paniagua Benito, donde la autora pretende establecer una relación entre los procesos cognitivos y la práctica de ajedrez, motivos por los cuales, este estudio demuestra que los procesos cognitivos son mejorados tras una práctica continua del ajedrez. Para la realización de esta investigación, se utilizó un diseño cuasi-experimental de pretest-postest con grupo de control no equivalente sobre una muestra de 60 alumnos de Educación Primaria pertenecientes a dos colegios públicos de la comunidad de Extremadura, de los que 30 alumnos participarán en un programa de ajedrez. Sobre esta muestra se aplicaron diversas pruebas y test que evaluaron la percepción, la atención, la memoria y las funciones ejecutivas de los alumnos participantes. Importante mencionar que se contó con la realización de cuestionarios y fichas de observación por parte de los docentes implicados. Finalmente, toda la información se recogió y analizó (mediante la prueba de Wilcoxon para los datos cuantitativos y la categorización de resul-

tados para los datos cualitativos) con el objetivos de conseguir, como resultado, probar de la práctica continuada del ajedrez mejora los procesos cognitivos.

En una línea más alejada pero dentro de la misma temática, podemos mencionar a [5], donde Sanango Peña, J.E pretende desarrollar una aplicación de visión artificial que, en conjunto con un motor de ajedrez y un brazo robótico, será capaz de interactuar con una persona en una partida de ajedrez en el mundo real. Para conseguir este objetivo, utiliza conocimientos en programación, tanto para el reconocimiento de piezas de ajedrez, jugadas y los movimientos que realizará el brazo robótico. Entre los objetivos de este proyecto se encuentran: el estudio de los principios de procesado de imágenes (con el fin de procesar las mismas para que sean tratadas por el ordenador), diseñar y desarrollar un módulo de visión artificial (que permita el reconocimiento del tablero de ajedrez, así como posiciones de las piezas), diseño de un elemento terminal que permita una adecuada sujeción de las piezas de ajedrez, etc

Todos las referencias mencionadas hasta ahora son principalmente trabajos de gente no muy reconocida en el campo (Trabajos de fin de grado, de máster,etc), pero también nos encontramos con ciertos artículos que merecen la pena mencionar. Ejemplo de ello es [6], el cual se centra en el desarrollo de un sistema de reconocimiento y seguimiento de piezas de ajedrez basado en aprendizaje profundo (deep learning), con el objetivo de analizar partidas de ajedrez. En concreto, los autores proponen un enfoque basado en redes neuronales convolucionales(CNN) para reconocer y seguir las piezas de ajedrez en tiempo real a partir de una secuencia de imágenes capturadas con diferentes configuraciones de cámara y condiciones de iluminación. Este sistema se integra con un motor de ajedrez que analiza los movimientos y genera una respuesta apropiada.

Artículo a mencionar también es [7], que describe el desarrollo de un sistema de reconocimiento de piezas de ajedrez en tiempo real, utilizando técnicas de aprendizaje profundo (deep learning). Su objetivo principal es el de crear un sistema capaz de reconocer todas las piezas de ajedrez en tiempo real, utilizando técnicas de aprendizaje profundo(deep learning). EL sistema propuesto utiliza una CNN para clasificar las imágenes de las piezas de ajedrez en función de su tipo. Además, se utiliza un algoritmo de seguimiento basado en el método de Kanade-Lucas-Tomasi (KLT) para identificar y rastrear el movimiento de las piezas en el tablero. Este tipo de sistemas pueden ser útiles en el análisis de partidas de ajedrez, la enseñanza y el entrenamiento de jugadores, así como en aplicaciones de ajedrez en línea.

En el penúltimo,[8] describe un enfoque híbrido de aprendizaje profundo para el reconocimiento de piezas de ajedrez en un tablero. Los autores combinan dos técnicas de detección de objetos: la detección de características basada en características y la detección de características basada en redes neuronales convolucionales para mejorar la precisión del reconocimiento de piezas. El enfoque híbrido se prueba en un conjunto de datos de imágenes de tableros de ajedrez, y se compara con otros métodos de aprendizaje profundo y métodos tradicionales de procesamiento de imágenes. Los resultados muestran que el enfoque híbrido logra una precisión de reconocimiento de piezas superior a los métodos

comparativos.

Finalmente, en [9] se describe un método para el reconocimiento automático de piezas de ajedrez utilizando una red neuronal convolucional mejorada llamada YOLOv3. Los autores mejoran la arquitectura de la red para aumentar la precisión y reducir el tiempo de procesamiento. El método se evalúa en un conjunto de datos de imágenes de tableros de ajedrez y se compara con otros métodos de aprendizaje profundo y métodos tradicionales de procesamiento de imágenes. Los resultados muestran que el método propuesto logra una precisión de reconocimiento de piezas superior a los métodos comparativos y puede procesar imágenes en tiempo real.

4. Desarrollo

4.1. Primera aproximación

4.1.1. Descripción

En esta primera aproximación se ha decidido resolver el subproblema de diferenciar entre imágenes de damas y peones usando 100 fotos, 50 de cada clase. Se usan fotografías de piezas en blanco y negro, aunque puede haber variaciones de color en las imágenes, esto siendo resultado de las diferentes calidades de las fotos realizadas y de la luz con la que si hicieron. Además puede haber dos tipos de fondos y las piezas pueden ser blancas o negras y variar ligeramente de forma. En este subproblema las pequeñas variaciones que se pueden dar en el contorno no afectan en la resolución porque, como explicamos a continuación, los atributos extraídos no tienen en cuenta el contorno de las piezas. En la Figura 2 se muestran algunas variaciones de las imágenes utilizadas.

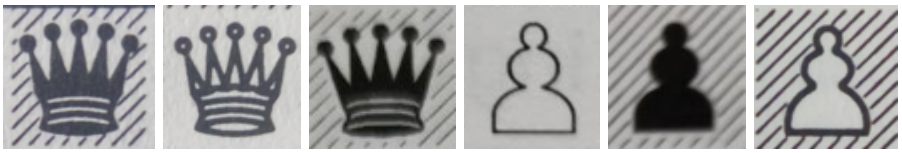

Figura 2: Distintas imágenes usadas en la primera aproximación

En este subproblema se ha decidido usar un único atributo: la relación entre la altura y la anchura de cada pieza, ya que los dos tipos de piezas tienen una relación ancho-alto claramente diferente, por lo que esta característica es diferencial. Para obtener este valor se ha calculado en cada imagen la bounding box de la pieza, que indica con un rectángulo lo que ocupa la pieza en la imagen. El valor final que se le pasa a los distintos modelos es simplemente la división entre ancho y alto de este rectángulo, por lo que no influyen las posibles variaciones

Figura 3: Cálculo de la bounding box de una pieza

en la forma del contorno. En la Figura 3 se muestra el proceso seguido para el cálculo de las bounding boxes.

En este proceso convertimos previamente la imagen a escala de grises para facilitar su procesamiento y, una vez calculadas la bounding box y la relación ancho-alto, normalizamos los valores obtenidos para pasárselos a los modelos.

Para realizar los experimentos, se utilizará la técnica de la validación cruzada, método utilizado para predicción, y que garantiza que los resultados son independientes de la partición entre datos de entrenamiento y test. En concreto, utilizaremos una aproximación de K iteraciones, siendo estas 10.

4.1.2. Resultados

En este apartado explicaremos las distintas características que empleamos con cada técnica y los resultados obtenidos tras sus ejecuciones. Estos resultados son el promedio de los tests de los 10 folds.

Resultados de Redes Neuronales								
	Configuraciones probadas							
Topología	[2]	[8]	[16]	[4, 2]	[8, 4]	[8, 4, 2]	[2, 8]	[16, 8]
	Valores obtenidos							
Precisión	0.9889	0.9889	0.9922	0.9889	0.9889	0.9889	0.9911	0.9900
Tasa de fallo	0.0111	0.0111	0.0078	0.0111	0.0111	0.0111	0.0089	0.0100
Sensibilidad	0.9857	0.9857	0.99	0.9857	0.9857	0.9857	0.9886	0.9871
Especificidad	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
PV+	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
PV-	0.9667	0.9667	0.9767	0.9667	0.9667	0.9667	0.9733	0.9700
F1-score	0.9923	0.9923	0.9946	0.9923	0.9923	0.9923	0.9938	0.9931

Tabla 1: Parámetros y métricas de evaluación de las redes neuronales.

Los siguientes parámetros se han utilizado con todas las topologías de redes neuronales:

- LearningRate = 0.01
- ValidationRatio = 0
- NumRepetitionsANNTraining = 10
- NumMaxEpochs = 400
- MaxEpochsVal = 6

Como se puede apreciar en la tabla anterior, para las redes neuronales se han obtenido los mejores resultados en la configuración que tiene una topología de 16 neuronas en la capa oculta ([16]). Además, el resto de parámetros utilizados es el mismo para todas las configuraciones (LearningRate, ValidationRatio, NumRepetitionsANNTraining, NumMaxEpochs, MaxEpochsVal).

Para esta, la precisión obtenida es del 99.22 %, siendo la más alta de todas las configuraciones probadas. Además, también presenta una tasa de fallo del 0.78 % y un F1-score del 99.46 %.

		Predicción	
		Dama	Peón
Real	Dama	3.70	0.00
	Peón	0.07	5.23

Tabla 2: Matriz de confusión obtenida con la mejor red neuronal.

Resultados de Árboles de Decisión						
	Configuraciones probadas					
MaxDepth	2	4	8	10	13	14
	Valores obtenidos					
Precisión	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Tasa de fallo	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Sensibilidad	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Especificidad	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
PV+	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
PV-	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
F1-score	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

Tabla 3: Parámetros y métricas de evaluación de los árboles de decisión.

Como se puede apreciar en la tabla anterior, para los árboles de decisión se han obtenido los mejores resultados en todas las configuraciones probadas de MaxDepth (2,4,8,10,13,14).

Para todas las configuraciones, la precisión obtenida es del 100 %. Además, también presenta una tasa de fallo del 0 % y un F1-score del 100 %.

		Predicción	
		Dama	Peón
Real	Dama	3.70	0.00
	Peón	0.00	5.30

Tabla 4: Matriz de confusión obtenida con todas las configuraciones probadas de Árboles de decisión.

Resultados de kNN						
	Configuraciones probadas					
MaxDepth	2	5	10	12	15	16
	Valores obtenidos					
Precisión	1.0000	0.9889	0.9889	0.9889	0.9889	0.9889
Tasa de fallo	0.0000	0.0111	0.0111	0.0111	0.0111	0.0111
Sensibilidad	1.0000	0.9857	0.9857	0.9857	0.9857	0.9857
Especificidad	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
PV+	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
PV-	1.0000	0.9667	0.9667	0.9667	0.9667	0.9667
F1-score	1.0000	0.9923	0.9923	0.9923	0.9923	0.9923

Tabla 5: Parámetros y métricas de evaluación de los árboles de decisión.

Como se puede apreciar en la tabla, para las diferentes configuraciones probadas con los valores de kNN (2,5,10,12,15,16), se han obtenido los mejores resultados con un NumNeighbors de 2, en el que se ha obtenido una precisión del 100 % y una tasa de fallo del 0.00 %. Además, el F1-score obtenido es del 100 %.

		Predicción	
		Dama	Peón
Real	Dama	3.70	0.00
	Peón	0.00	5.30

Tabla 6: Matriz de confusión obtenida con la mejor configuración de kNN.

Resultados de SVM								
	Configuraciones probadas							
Kernel	rbf	linear	poly	sigmoid	rbf	linear	poly	sigmoid
C	1	1	1	1	3	3	3	3
	Valores obtenidos							
Precisión	0.9889	0.9889	1.0000	0.9778	0.9889	0.9889	1.0000	0.9778
Tasa de fallo	0.0111	0.0111	0.0000	0.0222	0.0111	0.0111	0.0000	0.0222
Sensibilidad	0.9857	0.9857	1.0000	0.9524	0.9857	0.9857	1.0000	0.9524
Especificidad	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
PV+	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
PV-	0.9667	0.9667	1.0000	0.9524	0.9667	0.9667	1.0000	0.9524
F1-score	0.9923	0.9923	1.0000	0.9723	0.9923	0.9923	1.0000	0.9723

Tabla 7: Parámetros y métricas de evaluación de SVM.

Los siguientes parámetros se han utilizado con todas las configuraciones de SVM:

- KernelDegree = 3
- KernelGamma = 2

Como se puede apreciar en la tabla, para las distintas configuraciones probadas con los diferentes tipos de kernel (rbf,linear,poly,sigmoid), con valores de C de 1 y 3 para cada uno de ellos, se han obtenido los mejores resultados con el kernel poly, tanto en C valiendo 1 como valiendo 3. En los tres se han obtenido una precisión del 100 % y una tasa de fallo del 0.00 %. Además, el F1-score obtenido es del 100 %.

		Predicción	
		Dama	Peón
Real	Dama	3.70	0.00
	Peón	0.00	5.30

Tabla 8: Matriz de confusión obtenida con las mejores configuraciones de SVM.

4.1.3. Discusión

Las tablas anteriores muestran los valores obtenidos al aplicar cuatro algoritmos de clasificación (ANN,SVM,Árboles y kNN), usando la relación entre la altura y la anchura de la bounding box de la pieza como atributo.

Los resultados muestran que, en su mayoría, las técnicas de aprendizaje automático han sido capaces de identificar correctamente las piezas de ajedrez en las imágenes usando como aproximaciones la media y la desviación típica. Se puede identificar en las tablas de resultados de aproximaciones que, los algoritmos de Árboles de decisión, kNN y SVM han obtenido una precisión de un 100 %, mientras que el ANN ha obtenido una precisión del 99,22 %. La tasa de error fue de 0 para los algoritmos SVM, Árboles de decisión y kNN, y de 0.78 % para ANN.

Los valores de sensibilidad, especificidad, valor predictivo positivo, valor predictivo negativo y F1-score también fueron de 1 para SVM, Árboles de decisión y kNN, mientras que RNA presentó valores más bajos en todas estas métricas.

En cuanto a los algoritmos se refiere, se puede apreciar que los Árboles de decisión, kNN y SVM han funcionado mejor que el RNA. Esto puede deberse a que estos algoritmos son capaces de encontrar patrones más complejos en los datos. Además, la matriz de confusión de RNA ha tenido más dificultades para identificar correctamente algunas de las piezas.

Con esto, se puede llegar a la conclusión de que las características son lo suficientemente discriminativas, ya que hay técnicas que han llegado al 100 % de precisión, por lo que estos resultados sugieren que se tenga que ampliar el subproblema añadiendo nuevas figuras, ya que en esta no se puede mejorar más aunque se introduzcan más características.

4.2. Segunda aproximación

4.2.1. Descripción

Debido a los buenos resultados obtenidos en la previa implementación, la segunda aproximación se realizará, usando los mismos atributos, pero aplicándolas a un mayor número de piezas, así observando si estas características son suficientes, o si nos vemos en la necesidad de añadir más.

Para esta aproximación añadimos las imágenes con las figuras de alfiles y de caballos, manteniendo las características y atributos de la primera aproximación, pero ahora utilizando un total de 200 imágenes.

Al igual que en la anterior aproximación, se utilizará como atributo la relación alto-ancho de cada pieza, valor que se normaliza antes de pasarlo a los modelos. Para realizar los experimentos, se utilizará la técnica de la validación

cruzada, método utilizado para predicción, y que garantiza que los resultados son independientes de la partición entre datos de entrenamiento y test. En concreto, utilizaremos una aproximación de K iteraciones, siendo estas 10.

4.2.2. Resultados

En este apartado explicaremos las distintas características que empleamos con cada técnica y los resultados obtenidos tras sus ejecuciones. Estos resultados son el promedio de los tests de los 10 folds.

De gran importancia mencionar que los siguientes parámetros se han utilizado con todas las topologías de redes neuronales:

- LearningRate = 0.01
- ValidationRatio = 0
- NumRepetitionsANNTraining = 10
- NumMaxEpochs = 400
- MaxEpochsVal = 6

Resultados de Redes Neuronales								
	Configuraciones probadas							
Topología	[2]	[8]	[16]	[4, 2]	[8, 4]	[8, 4, 2]	[2, 8]	[16, 8]
	Valores obtenidos							
Precisión	0.5996	0.7095	0.7099	0.7212	0.7205	0.6535	0.7128	0.6746
Tasa de fallo	0.4004	0.2905	0.2901	0.2788	0.2795	0.3465	0.2872	0.3254
Sensibilidad	0.5996	0.7095	0.7099	0.7212	0.7205	0.6535	0.7128	0.6746
Especificidad	0.8360	0.8899	0.8880	0.8942	0.8929	0.8632	0.8907	0.8741
PV+	0.5036	0.5669	0.5639	0.5981	0.5959	0.5135	0.5670	0.5514
PV-	0.8851	0.921	0.9207	0.9228	0.9232	0.9070	0.9217	0.9076
F1-score	0.5156	0.6223	0.6217	0.6415	0.6403	0.5583	0.6250	0.5934

Tabla 9: Parámetros y métricas de evaluación de las redes neuronales.

Como se puede apreciar en la tabla anterior, para las redes neuronales se han obtenido los mejores resultados en la configuración que tiene una topología de [4,2] neuronas. Además, el resto de parámetros utilizados es el mismo para todas las configuraciones (LearningRate, ValidationRatio, NumRepetitionsANNTraining, NumMaxEpochs, MaxEpochsVal). Para esta, la precisión obtenida es del 72.12 %, siendo la más alta de todas las configuraciones probadas. Además, también presenta una tasa de fallo del 27.88 % y un F1-score del 64.15 %.

		Predicción			
		Dama	Peón	Caballo	Alfil
Real	Dama	4.71	0.13	0.00	0.46
	Peón	0.00	3.70	0.00	0.00
	Caballo	0.83	0.08	0.22	2.17
	Alfil	0.34	0.01	0.00	4.35

Tabla 10: Matriz de confusión obtenida con la mejor red neuronal.

Resultados de Árboles de Decisión						
	Configuraciones probadas					
MaxDepth	2	4	8	10	13	14
	Valores obtenidos					
Precisión	0.7064	0.8383	0.8328	0.8390	0.8390	0.8390
Tasa de fallo	0.2936	0.1617	0.1672	0.1610	0.1610	0.1610
Sensibilidad	0.7064	0.8383	0.8328	0.8390	0.8390	0.8390
Especificidad	0.8935	0.9404	0.9399	0.9420	0.9420	0.9420
PV+	0.6004	0.8669	0.8524	0.8574	0.8574	0.8574
PV-	0.9207	0.9479	0.9431	0.9457	0.9457	0.9457
F1-score	0.6309	0.8321	0.8271	0.8334	0.8334	0.8334

Tabla 11: Parámetros y métricas de evaluación de los árboles de decisión.

La tabla anterior presenta los resultados de evaluación de diferentes configuraciones de árboles de decisión, donde se varió el parámetro de MaxDepth. Se puede observar que para las configuraciones de 10, 13 y 14 en el valor de MaxDepth, se ha obtenido la precisión más alta, siendo esta del 83.90 %. Además, se ha obtenido una tasa de fallo en ellas del 16.10 % y un F1-score del 83.34 %.

		Predicción			
		Dama	Peón	Caballo	Alfil
Real	Dama	4.30	0.00	0.60	0.40
	Peón	0.00	3.70	0.00	0.00
	Caballo	0.30	0.00	3.10	0.90
	Alfil	0.40	0.00	0.30	4.00

Tabla 12: Matriz de confusión obtenida con las mejores configuraciones de árboles de decisión.

Resultados de kNN						
	Configuraciones probadas					
MaxDepth	2	5	10	12	15	16
	Valores obtenidos					
Precisión	0.7875	0.8675	0.8226	0.8226	0.8014	0.8164
Tasa de fallo	0.2125	0.1325	0.1774	0.1774	0.1986	0.1836
Sensibilidad	0.7875	0.8675	0.8226	0.8226	0.8014	0.8164
Especificidad	0.9260	0.9520	0.9365	0.9369	0.9298	0.9348
PV+	0.7888	0.8797	0.8289	0.8298	0.8149	0.8283
PV-	0.9275	0.9559	0.9436	0.9435	0.9345	0.9399
F1-score	0.7786	0.8641	0.8102	0.8110	0.7938	0.8077

Tabla 13: Parámetros y métricas de evaluación de kNN.

Como se puede apreciar en la tabla, para las diferentes configuraciones probadas con los valores de kNN (2,5,10,12,15,16), se han obtenido los mejores resultados con un NumNeighbors de 5, en el que se ha obtenido una precisión del 86,75 % y una tasa de fallo del 13.25 %. Además, el F1-score obtenido es del 86.41 %.

		Predicción			
		Dama	Peón	Caballo	Alfil
Real	Dama	4.40	0.10	0.40	0.40
	Peón	0.00	3.70	0.00	0.00
	Caballo	0.30	0.00	3.30	0.70
	Alfil	0.30	0.00	0.20	4.20

Tabla 14: Matriz de confusión obtenida con la mejor configuración de kNN.

Resultados de SVM								
	Configuraciones probadas							
Kernel	rbf	linear	poly	sigmoid	rbf	linear	poly	sigmoid
C	1	1	1	1	3	3	3	3
	Valores obtenidos							
Precisión	0.7114	0.4944	0.4944	0.5467	0.7008	0.7114	0.5794	0.6885
Tasa de fallo	0.2886	0.5056	0.5056	0.4533	0.2992	0.2886	0.4206	0.3115
Sensibilidad	0.7114	0.4944	0.4944	0.5467	0.7008	0.7114	0.5794	0.6885
Especificidad	0.8915	0.7895	0.7889	0.8134	0.8910	0.8915	0.8243	0.8844
PV+	0.5690	0.3096	0.3140	0.4894	0.5919	0.5672	0.5052	0.5530
PV-	0.9217	0.8685	0.8680	0.8740	0.9183	0.9215	0.8822	0.9147
F1-score	0.6242	0.3608	0.3633	0.4505	0.6248	0.6237	0.4956	0.6032

Tabla 15: Parámetros y métricas de evaluación de SVM.

Los siguientes parámetros se han utilizado con todas las configuraciones de SVM:

- KernelDegree = 3
- KernelGamma = 2

Como se puede apreciar en la tabla, para las distintas configuraciones probadas con los diferentes tipos de kernel (rbf,linear,poly,sigmoid), con valores de C de 1 y 3 para cada uno de ellos, se han obtenido los mejores resultados con el kernel rbf con C 1 y en kernel linear con C 3. En ambos se ha obtenido una precisión del 71.14 % y una tasa de fallo del 28.86 %. Además, el F1-score obtenido es del 62.42 % en rbf y del 62.37 % en linear. Cabe destacar que para todas las configuraciones se han utilizado los parámetros KernelDegree (con valor 3) y KernelGamma(con valor 2).

		Predicción			
		Dama	Peón	Caballo	Alfil
Real	Dama	4.70	0.10	0.00	0.50
	Peón	0.00	3.70	0.00	0.00
	Caballo	1.70	0.00	0.00	2.60
	Alfil	0.30	0.00	0.00	4.40

Tabla 16: Matriz de confusión obtenida con las mejores configuraciones de SVM.

4.2.3. Discusión

Después de añadir los dos nuevos tipos de piezas (alfil y caballo), la calidad de los resultados disminuyó sustancialmente. Dos de los algoritmos, ANN y SVM no dan resultados aceptables, sus mejores resultados no alcanzan una precisión del 80 %. Mientras que los otros dos, árboles de decisión y kNN sí que dan resultados que se podrían considerar aceptables, aunque ninguno supera una precisión del 90 %. Teniendo en cuenta la simplicidad de los atributos usados (solo se usa la relación altura/anchura) y que la calidad de los resultados no alcanza los requisitos necesarios, en la siguiente aproximación se mantendrá el número de piezas añadiendo nuevos atributos.

Estos resultados eran esperables, puesto que la única característica que estamos usando es la relación entre altura y anchura de las piezas. Si bien con este atributo es fácil diferenciar entre damas y peones, las dos nuevas piezas no varían demasiado en este aspecto, teniendo una relación altura/anchura similar a la de la reina. Cabe destacar que los algoritmos tienen más problemas diferenciando las imágenes del caballo, ya que su relación anchura/altura es prácticamente igual a 1.

4.3. Tercera aproximación

4.3.1. Descripción

Debido a los resultados obtenidos en la previa implementación, la tercera aproximación se realizará, usando las mismas piezas, pero añadiéndole un nuevo atributo, siendo este la relación entre el número de píxeles del contorno de la parte superior y de la parte inferior de la bounding box de la imagen.

Para obtener este valor se ha calculado la bounding box y se ha dividido horizontalmente en 5 partes iguales, de las que se han usado la superior y la inferior. En cada parte el contorno se obtiene con el primer píxel de pieza que se encuentra en cada columna de la imagen, y se cuenta la cantidad de píxeles encontrados. Finalmente se calcula la división entre el número de píxeles encontrados en las partes superior e inferior, y se normaliza este valor antes de

Figura 4: Cálculo del contorno de una pieza

pasárselo al modelo. En la Figura 4 se muestra el contorno calculado para una imagen de un alfil.

Por lo tanto, usaremos las mismas imágenes que en la aproximación anterior, pero calculando este nuevo atributo, con el cual esperamos que la precisión de nuestras técnicas aumente.

Para realizar los experimentos, se utilizará la técnica de la validación cruzada, método utilizado para predicción, y que garantiza que los resultados son independientes de la partición entre datos de entrenamiento y test. En concreto, utilizaremos una aproximación de K iteraciones, siendo estas 10.

4.3.2. Resultados

En este apartado explicaremos las distintas características que empleamos con cada técnica y los resultados obtenidos tras sus ejecuciones. Estos resultados son el promedio de los tests de los 10 folds.

Resultados de Redes Neuronales								
	Configuraciones probadas							
Topología	[2]	[8]	[16]	[4, 2]	[8, 4]	[8, 4, 2]	[2, 8]	[16, 8]
	Valores obtenidos							
Precisión	0.7095	0.9240	0.9139	0.8046	0.9250	0.7287	0.8364	0.9127
Tasa de fallo	0.2905	0.0760	0.0861	0.1954	0.0750	0.2713	0.1636	0.0873
Sensibilidad	0.7095	0.9240	0.9139	0.8046	0.9250	0.7287	0.8364	0.9127
Especificidad	0.8833	0.9748	0.9700	0.9238	0.9760	0.8929	0.9380	0.9701
PV+	0.6062	0.9362	0.9273	0.7631	0.9382	0.6388	0.8252	0.9245
PV-	0.9241	0.9737	0.9708	0.9455	0.9738	0.9277	0.9504	0.9706
F1-score	0.6310	0.9240	0.9132	0.7641	0.9249	0.6607	0.8163	0.9105

Tabla 17: Parámetros y métricas de evaluación de las redes neuronales.

Como se puede apreciar en la tabla anterior, para las redes neuronales se han obtenido los mejores resultados en la configuración que tiene una topología de [8,4]. Para esta, la precisión obtenida es del 92.5 %, siendo la más alta de todas las configuraciones probadas. Además, también presenta una tasa de fallo del 7.5 % y un F1-score del 92.49 %.

		Predicción			
		Dama	Peón	Caballo	Alfil
Real	Dama	4.55	0.11	0.55	0.09
	Peón	0.00	3.70	0.00	0.72
	Caballo	0.07	0.03	4.02	0.18
	Alfil	0.00	0.01	0.30	4.39

Tabla 18: Matriz de confusión obtenida con la mejor red neuronal.

Resultados de Árboles de Decisión						
	Configuraciones probadas					
MaxDepth	2	4	8	10	13	14
	Valores obtenidos					
Precisión	0.7704	0.8840	0.9447	0.9447	0.9447	0.9447
Tasa de fallo	0.2296	0.1160	0.0553	0.0553	0.0553	0.0553
Sensibilidad	0.7704	0.8840	0.9447	0.9447	0.9447	0.9447
Especificidad	0.9042	0.9608	0.9808	0.9808	0.9808	0.9808
PV+	0.6768	0.8948	0.9537	0.9537	0.9537	0.9537
PV-	0.9411	0.9597	0.9808	0.9808	0.9808	0.9808
F1-score	0.7037	0.8833	0.9443	0.9443	0.9443	0.9443

Tabla 19: Parámetros y métricas de evaluación de los árboles de decisión.

La tabla anterior presenta los resultados de evaluación de diferentes configuraciones de árboles de decisión, donde se varió el parámetro de MaxDepth. Se puede observar que para las configuraciones de 8, 10, 13 y 14 en el valor de MaxDepth, se ha obtenido la precisión más alta, siendo esta del 94.47 %. Además, se ha obtenido una tasa de fallo en ellas del 5.53 % y un F1-score del 94.43 %.

		Predicción			
		Dama	Peón	Caballo	Alfil
Real	Dama	4.80	0.00	0.40	0.10
	Peón	0.00	3.70	0.00	0.00
	Caballo	0.10	0.00	4.10	0.10
	Alfil	0.20	0.00	0.10	4.40

Tabla 20: Matriz de confusión obtenida con las mejores configuraciones de árboles de decisión.

Resultados de kNN						
	Configuraciones probadas					
MaxDepth	2	5	10	12	15	16
	Valores obtenidos					
Precisión	0.9329	0.9329	0.9274	0.9274	0.9274	0.9274
Tasa de fallo	0.0671	0.0671	0.0726	0.0726	0.0726	0.0726
Sensibilidad	0.9329	0.9329	0.9274	0.9274	0.9274	0.9274
Especificidad	0.9781	0.9781	0.9771	0.9775	0.9775	0.9775
PV+	0.9422	0.9422	0.9403	0.9405	0.9405	0.9405
PV-	0.9762	0.9762	0.9746	0.9746	0.9746	0.9746
F1-score	0.9327	0.9327	0.9269	0.9269	0.9269	0.9269

Tabla 21: Parámetros y métricas de evaluación de kNN.

Como se puede apreciar en la tabla, para las diferentes configuraciones probadas con los valores de kNN (2,5,10,12,15,16), se han obtenido los mejores resultados con un NumNeighbors de 2 y 5, en el que se ha obtenido una precisión del 93.29 % y una tasa de fallo del 6.71 %. Además, el F1-score obtenido es del 93.27 %.

		Predicción			
		Dama	Peón	Caballo	Alfil
Real	Dama	4.60	0.00	0.60	0.10
	Peón	0.00	3.70	0.00	0.00
	Caballo	0.00	0.00	4.10	0.20
	Alfil	0.10	0.00	0.20	4.40

Tabla 22: Matriz de confusión obtenida con las mejores configuraciones de kNN.

Resultados de SVM								
	Configuraciones probadas							
Kernel	rbf	linear	poly	sigmoid	rbf	linear	poly	sigmoid
C	1	1	1	1	3	3	3	3
	Valores obtenidos							
Precisión	0.9011	0.7337	0.7337	0.7331	0.9329	0.7554	0.7388	0.8943
Tasa de fallo	0.0989	0.2662	0.2662	0.2669	0.0671	0.2446	0.2612	0.1057
Sensibilidad	0.9011	0.7337	0.7337	0.7331	0.9329	0.7554	0.7388	0.8943
Especificidad	0.9641	0.8909	0.8892	0.8959	0.9787	0.9014	0.8921	0.9656
PV+	0.9155	0.6092	0.6103	0.6832	0.9448	0.7187	0.6355	0.9155
PV-	0.9673	0.9312	0.9315	0.9243	0.9763	0.9298	0.9322	0.9666
F1-score	0.8995	0.6521	0.6518	0.6800	0.9332	0.7088	0.6616	0.8930

Tabla 23: Parámetros y métricas de evaluación de SVM.

Los siguientes parámetros se han utilizado con todas las configuraciones de SVM:

- KernelDegree = 3
- KernelGamma = 2

Como se puede apreciar en la tabla, para las distintas configuraciones probadas con los diferentes tipos de kernel (rbf,linear,poly,sigmoid), con valores de C de 1 y 3 para cada uno de ellos, se han obtenido los mejores resultados con el kernel rbf con C 3. En este se ha obtenido una precisión del 93.29 % y una tasa de fallo del 6.71 %. Además, el F1-score obtenido es del 93.32 %.

		Predicción			
		Dama	Peón	Caballo	Alfil
Real	Dama	4.60	0.10	0.50	0.10
	Peón	0.00	3.70	0.00	0.00
	Caballo	0.00	0.00	4.10	0.20
	Alfil	0.00	0.00	0.30	4.40

Tabla 24: Matriz de confusión obtenida con la mejor configuración de SVM.

4.3.3. Discusión

Tras la ejecución de esta aproximación, con la adición del nuevo atributo, las diferentes arquitecturas reciben una mejoría en su desempeño. Para las ANN, se obtuvo una precisión del 92,5 %, teniendo esta una topología de [8,4]. Para los árboles de decisión, se obtuvo una precisión del 94,47 %, teniendo 8 como profundidad máxima, siendo esta la técnica con la precisión más alta. En los k-vecinos, se obtuvo una precisión del 93,29 %, con un número de vecinos de 2. Por otra parte, en los SVM, se obtiene una precisión del 93,29 %, usando el kernel rbfz un valor de C de 3, siendo esta la peor técnica.

Analizando estos resultados, creemos que la precisión es satisfactoria, por lo tanto, en la siguiente aproximación, procederemos a probar con los mismos atributos, pero con la adición de una nueva pieza (figura de torre).

4.4. Cuarta aproximación

4.4.1. Descripción

En vista de los buenos resultados obtenidos en la aproximación anterior, decidimos añadir una nueva pieza pero conservando las características ya utilizadas, para comprobar si estas son suficientes para clasificar correctamente esta pieza junto a las añadidas previamente. Para esto hemos sumado a la base de datos 50 nuevas imágenes, en este caso con torres.

Al igual que en la anterior aproximación, se utilizarán como atributos las relaciones entre alto y ancho y entre contorno superior e inferior de cada pieza, los cuales se normalizan antes de pasarse a los modelos. Para realizar los experimentos, se utilizará la técnica de la validación cruzada, método utilizado para predicción, y que garantiza que los resultados son independientes de la partición entre datos de entrenamiento y test. En concreto, utilizaremos una aproximación de K iteraciones, siendo estas 10.

4.4.2. Resultados

En este apartado explicaremos las distintas características que empleamos con cada técnica y los resultados obtenidos tras sus ejecuciones. Estos resultados son el promedio de los tests de los 10 folds.

Resultados de Redes Neuronales								
	Configuraciones probadas							
Topología	[2]	[8]	[16]	[4, 2]	[8, 4]	[8, 4, 2]	[2, 8]	[16, 8]
	Valores obtenidos							
Precisión	0.6964	0.8892	0.8668	0.6924	0.8870	0.6202	0.7958	0.8765
Tasa de fallo	0.3036	0.1108	0.1332	0.3076	0.1130	0.3798	0.2042	0.1235
Sensibilidad	0.6964	0.8892	0.8668	0.6924	0.8870	0.6203	0.7958	0.8765
Especificidad	0.9084	0.9690	0.9620	0.9104	0.9698	0.8914	0.9422	0.9654
PV+	0.5874	0.9061	0.8832	0.6186	0.8939	0.5240	0.7992	0.8873
PV-	0.9386	0.9730	0.9681	0.9365	0.9729	0.9218	0.9542	0.9703
F1-score	0.6154	0.8859	0.8598	0.6240	0.8807	0.5378	0.7742	0.8688

Tabla 25: Parámetros y métricas de evaluación de las redes neuronales.

Como se puede apreciar en la tabla anterior, para las redes neuronales se han obtenido los mejores resultados en la configuración que tiene una topología de 8 neuronas. Para esta, la precisión obtenida es del 88.92 %, siendo la más alta de todas las configuraciones probadas. Además, también presenta una tasa de fallo del 11.08 % y un F1-score del 88.59 %. Interesante mencionar que con una topología de [8,4] el valor se acerca mucho al mejor resultado obtenido, habiendo solo una diferencia de 0.22 % entre sus valores de precisión y de tasas de fallo, y de 0.55 % en el F1-score.

Los siguientes parámetros se han utilizado con todas las topologías de redes neuronales:

- LearningRate = 0.01
- ValidationRatio = 0
- NumRepetitionsANNTraining = 10
- NumMaxEpochs = 400
- MaxEpochsVal = 6

		Predicción				
		Dama	Peón	Caballo	Alfil	Torre
Real	Dama	4.76	0.10	0.34	0.10	0.00
	Peón	0.00	3.70	0.00	0.00	0.00
	Caballo	0.62	0.00	3.38	0.20	0.10
	Alfil	0.00	0.00	0.30	4.40	0.00
	Torre	0.42	0.01	0.18	0.10	3.29

Tabla 26: Matriz de confusión obtenida con la mejor red neuronal.

Resultados de Árboles de Decisión						
	Configuraciones probadas					
MaxDepth	2	4	8	10	13	14
	Valores obtenidos					
Precisión	0.5992	0.8698	0.8955	0.9050	0.9092	0.9092
Tasa de fallo	0.4008	0.1302	0.1045	0.0950	0.0908	0.0908
Sensibilidad	0.5992	0.8698	0.8955	0.9050	0.9092	0.9092
Especificidad	0.8730	0.9665	0.9726	0.9751	0.9762	0.9762
PV+	0.4630	0.8879	0.9115	0.9225	0.9267	0.9267
PV-	0.9223	0.9667	0.9742	0.9762	0.9776	0.9776
F1-score	0.4956	0.8663	0.8912	0.9010	0.9051	0.9051

Tabla 27: Parámetros y métricas de evaluación de los árboles de decisión.

La tabla anterior presenta los resultados de evaluación de diferentes configuraciones de árboles de decisión, donde se varió el parámetro de MaxDepth. Se puede observar que para las configuraciones de 13 y 14 en el valor de MaxDepth, se ha obtenido la precisión más alta, siendo esta del 90.92 %. Además, se ha obtenido una tasa de fallo en ellas del 9.08 % y un F1-score del 90.51 %. Cabe destacar que para una topología de 10, el valor es muy próximo al valor óptimo de esta tabla, presentado una diferencia ínfima de 0.42 % entre dichas topologías.

		Predicción				
		Dama	Peón	Caballo	Alfil	Torre
Real	Dama	4.70	0.00	0.50	0.10	0.00
	Peón	0.00	3.70	0.00	0.00	0.00
	Caballo	0.10	0.00	4.00	0.10	0.10
	Alfil	0.10	0.00	0.20	4.20	0.20
	Torre	0.30	0.10	0.10	0.10	3.40

Tabla 28: Matriz de confusión obtenida con la mejor configuración de árboles de decisión.

Resultados de kNN						
	Configuraciones probadas					
MaxDepth	2	5	10	12	15	16
	Valores obtenidos					
Precisión	0.9103	0.9103	0.9148	0.9103	0.9058	0.9103
Tasa de fallo	0.0897	0.0897	0.0852	0.0897	0.0942	0.0897
Sensibilidad	0.9103	0.9103	0.9148	0.9103	0.9058	0.9103
Especificidad	0.9759	0.9762	0.9776	0.9768	0.9761	0.9768
PV+	0.9208	0.9200	0.9246	0.9223	0.9197	0.9223
PV-	0.9774	0.9774	0.9788	0.9775	0.9764	0.9775
F1-score	0.9086	0.9083	0.9129	0.9088	0.9040	0.9088

Tabla 29: Parámetros y métricas de evaluación kNN.

Como se puede apreciar en la tabla, para las diferentes configuraciones probadas con los valores de kNN (2,5,10,12,15,16), se han obtenido los mejores resultados con un NumNeighbors de 10, en el que se ha obtenido una precisión del 91.48 % y una tasa de fallo del 8.52 %. Además, el F1-score obtenido es del 90.83 %.

		Predicción				
		Dama	Peón	Caballo	Alfil	Torre
Real	Dama	4.60	0.10	0.50	0.10	0.00
	Peón	0.00	3.70	0.00	0.00	0.00
	Caballo	0.00	0.00	4.00	0.20	0.10
	Alfil	0.00	0.00	0.20	4.50	0.00
	Torre	0.40	0.00	0.20	0.10	3.30

Tabla 30: Matriz de confusión obtenida con las mejores configuraciones de kNN.

Resultados de SVM								
	Configuraciones probadas							
Kernel	rbf	linear	poly	sigmoid	rbf	linear	poly	sigmoid
C	1	1	1	1	3	3	3	3
	Valores obtenidos							
Precisión	0.8745	0.7516	0.7516	0.6714	0.9103	0.7652	0.7516	0.8081
Tasa de fallo	0.1255	0.2484	0.2484	0.3286	0.0897	0.2348	0.2484	0.1919
Sensibilidad	0.8745	0.7516	0.7516	0.6714	0.9103	0.7652	0.7516	0.8081
Especificidad	0.9644	0.9229	0.9222	0.9133	0.9768	0.9281	0.9224	0.9523
PV+	0.8935	0.6653	0.6655	0.6693	0.9232	0.7319	0.6651	0.8335
PV-	0.9697	0.9484	0.9485	0.9281	0.9775	0.9476	0.9484	0.9573
F1-score	0.8704	0.6877	0.6875	0.6257	0.9091	0.7257	0.6876	0.8011

Tabla 31: Parámetros y métricas de evaluación de SVM.

Los siguientes parámetros se han utilizado con todas las configuraciones de SVM:

- KernelDegree = 3
- KernelGamma = 2

Como se puede apreciar en la tabla, para las distintas configuraciones probadas con los diferentes tipos de kernel (rbf,linear,poly,sigmoid), con valores de C de 1 y 3 para cada uno de ellos, se han obtenido los mejores resultados con el kernel rbf con C 3. En este se ha obtenido una precisión del 91.03 % y una tasa de fallo del 08.97 %. Además, el F1-score obtenido es del 90.91 %. Cabe destacar que para todas las configuraciones se han utilizado los parámetros KernelDegree (con valor 3) y KernelGamma(con valor 2).

		Predicción				
		Dama	Peón	Caballo	Alfil	Torre
Real	Dama	4.60	0.10	0.50	0.10	0.00
	Peón	0.00	3.70	0.00	0.00	0.00
	Caballo	0.00	0.00	4.00	0.20	0.10
	Alfil	0.00	0.00	0.30	4.40	0.00
	Torre	0.30	0.00	0.30	0.10	3.30

Tabla 32: Mejor matriz de confusión para SVM.

4.4.3. Discusión

Tras la ejecución de esta aproximación, con la adición de la nueva pieza, las diferentes arquitecturas reciben un ligero descenso en su desempeño. Para las ANN, se obtuvo una precisión del 88.92 %, teniendo esta una topología de 8 neuronas, siendo esta la peor técnica. Para los árboles de decisión, se obtuvo una precisión del 90.92 %, teniendo 13 y 14 como profundidad máxima. En los k-vecinos, se obtuvo una precisión del 91.48 %, con un número de vecinos de 10, siendo esta la técnica con la precisión más alta. Por otra parte, en los SVM, se obtiene una precisión del 91.03 %, usando el kernel rbf y un valor de C de 3.

Analizando estos resultados, creemos que la precisión es satisfactoria, por lo tanto, en la siguiente aproximación, procederemos a probar con los mismos atributos, pero con la adición de una nueva pieza (figura de rey).

4.5. Quinta aproximación

4.5.1. Descripción

Dados los prometedores resultados obtenidos en la anterior aproximación, se ha decidido realizar una nueva prueba para evaluar si las características utilizadas hasta ahora son suficientes para clasificar correctamente una nueva pieza que hemos añadido a nuestro conjunto de datos. Para esto, añadimos la figura del rey.

Al igual que en el enfoque anterior, utilizaremos las relaciones entre el alto y el ancho, y entre el contorno superior e inferior de cada pieza como atributos. Estos atributos se normalizan antes de ser utilizados en los modelos de clasificación. Para llevar a cabo los experimentos, utilizaremos la técnica de la validación cruzada, la cual es utilizada para predecir y garantizar que los resultados sean independientes de la partición de los datos de entrenamiento y prueba.

Con esto, buscamos evaluar la capacidad de nuestro enfoque actual para clasificar correctamente la nueva pieza introducida, utilizando un conjunto de datos ampliado y una técnica robusta de evaluación. Los resultados obtenidos

en esta prueba nos permitirán determianr si nuestras características actuales son suficientes o si se requiere algún ajuste adiccional para mejorar la precisión de la clasificación.

4.5.2. Resultados

En este apartado explicaremos las distintas características que empleamos con cada técnica y los resultados obtenidos tras sus ejecuciones. Estos resultados son el promedio de los tests de los 10 folds.

Resultados de Redes Neuronales								
	Configuraciones probadas							
Topología	[2]	[8]	[16]	[4, 2]	[8, 4]	[8, 4, 2]	[2, 8]	[16, 8]
	Valores obtenidos							
Precisión	0.6612	0.7939	0.7961	0.6083	0.7528	0.5364	0.6853	0.7743
Tasa de fallo	0.3388	0.2061	0.2039	0.3917	0.2472	0.4636	0.3147	0.2257
Sensibilidad	0.6612	0.7939	0.7961	0.6083	0.7528	0.5364	0.6853	0.7743
Especificidad	0.9215	0.9533	0.9536	0.9116	0.9439	0.8959	0.9294	0.9485
PV+	0.5929	0.8107	0.8131	0.5282	0.7488	0.4298	0.6707	0.7805
PV-	0.9406	0.9605	0.9608	0.931	0.9532	0.9208	0.9422	0.9573
F1-score	0.5917	0.7761	0.7781	0.5319	0.7248	0.4409	0.6458	0.7499

Tabla 33: Parámetros y métricas de evaluación de las redes neuronales.

Si analizamos los resultamos mostrados en la tabla anterior, podemos obaservar que la configuración con topología [16] obtuvo mayoro preción, sensibilidad y especificidad, lo que indica una mayor capacidad para clasificar correctamente los datos. Por otro lado, la configuración [8,4,2] mostró la menor precisión y sensibilidad, lo que sugiere un rendimiento menor en cuanto a la clasificación.

Cabe mencionar que los valores de las tasas de fallo, PV+ y PV- varían dependiendo de la configuración de la red neuronal, pero en general, se puede apreciar una tendencia a obtener tasas de fallo más bajas y valores predictivos más altos a medida que aumenta la complejidad de la red neuronal.

Los siguientes parámetros se han utilizado con todas las topologías de redes neuronales:

- LearningRate = 0.01
- ValidationRatio = 0

- NumRepetitionsANNTraining = 10
- NumMaxEpochs = 400
- MaxEpochsVal = 6

		Predicción					
		Dama	Peón	Caballo	Alfil	Torre	Rey
Real	Dama	4.79	0.10	0.29	0.00	0.02	0.10
	Peón	0.00	3.70	0.00	0.00	0.00	0.00
	Caballo	1.54	0.00	2.46	0.04	0.10	0.16
	Alfil	0.04	0.00	0.06	4.18	0.00	0.42
	Torre	0.44	0.00	0.16	0.00	3.30	0.10
	Rey	0.75	0.00	0.22	0.82	0.10	2.91

Tabla 34: Mejor matriz de confusión para redes neuronales.

Resultados de SVM								
	Configuraciones probadas							
Kernel	rbf	linear	poly	sigmoid	rbf	linear	poly	sigmoid
C	1	1	1	1	3	3	3	3
	Valores obtenidos							
Precisión	0.8295	0.6749	0.7322	0.5510	0.8549	0.7285	0.7285	0.6373
Tasa de fallo	0.1705	0.3251	0.2678	0.4490	0.1451	0.2715	0.2715	0.3627
Sensibilidad	0.8295	0.6749	0.7322	0.5510	0.8549	0.7285	0.7285	0.6373
Especificidad	0.9617	0.9238	0.9349	0.9056	0.9689	0.9358	0.9345	0.9258
PV+	0.8491	0.6192	0.6819	0.4627	0.8632	0.7139	0.6797	0.5694
PV-	0.9658	0.9414	0.9519	0.9198	0.9702	0.9488	0.951	0.9323
F1-score	0.8239	0.6122	0.6839	0.4577	0.8494	0.6965	0.6807	0.5822

Tabla 35: Parámetros y métricas de evaluación de SVM.

Los siguientes parámetros se han utilizado con todas las configuraciones de SVM:

- KernelDegree = 3
- KernelGamma = 2

Como se puede apreciar en la tabla, para las distintas configuraciones probadas con los diferentes tipos de kernel (rbf,linear,poly,sigmoid), con valores de C de 1 y 3 para cada uno de ellos, se han obtenido los mejores resultados con el kernel rbf con C 3. En este se ha obtenido una precisión del 85.49 % y una tasa de fallo del 14.51 %. Además, el F1-score obtenido es del 84.94 %.

		Predicción					
		Dama	Peón	Caballo	Alfil	Torre	Rey
Real	Dama	4.60	0.10	0.50	0.00	0.00	0.10
	Peón	0.00	3.70	0.00	0.00	0.00	0.00
	Caballo	0.00	0.00	4.00	0.10	0.10	0.10
	Alfil	0.00	0.00	0.10	4.00	0.00	0.60
	Torre	0.30	0.00	0.30	0.00	3.30	0.10
	Rey	0.70	0.00	0.40	0.30	0.10	3.30

Tabla 36: Mejor matriz de confusión para SVM.

Resultados de Árboles de Decisión						
	Configuraciones probadas					
MaxDepth	2	4	8	10	13	14
	Valores obtenidos					
Precisión	0.4773	0.838	0.8742	0.8779	0.8779	0.8779
Tasa de fallo	0.5227	0.1620	0.1258	0.1221	0.1221	0.1221
Sensibilidad	0.4773	0.8380	0.8742	0.8779	0.8779	0.8779
Especificidad	0.8771	0.9642	0.9737	0.9743	0.9743	0.9743
PV+	0.3129	0.8425	0.8790	0.8845	0.8851	0.8851
PV-	0.9100	0.9678	0.9746	0.9755	0.9755	0.9755
F1-score	0.3553	0.8269	0.8695	0.8729	0.8733	0.8733

Tabla 37: Parámetros y métricas de evaluación de los árboles de decisión.

La tabla anterior muestra los resultados de evaluación de diferentes configuraciones de árboles de decisión, donde se varió el parámetro de MaxDepth. Como se puede apreciar en dicha tabla, las configuraciones con una profundidad máxima de 10, 13 y 14 obtuvieron las mayores precisiones (87.79 %), sensibilidades (87.79 %) y especificidades (97.43 %), lo cual demuestra una mayor capacidad para clasificar correctamente los datos frente a las profundidades más bajas, las cuales obtuvieron peores precisiones (47.73 %) la de profundidad 2.

		Predicción					
		Dama	Peón	Caballo	Alfil	Torre	Rey
Real	Dama	4.60	0.00	0.20	0.20	0.00	0.30
	Peón	0.00	3.70	0.00	0.00	0.00	0.00
	Caballo	0.10	0.00	3.90	0.20	0.10	0.00
	Alfil	0.10	0.00	0.10	4.30	0.00	0.20
	Torre	0.10	0.10	0.10	0.00	3.30	0.40
	Rey	0.40	0.00	0.10	0.20	0.40	3.70

Tabla 38: Mejor matriz de confusión para árboles de decisión.

Resultados de kNN						
	Configuraciones probadas					
MaxDepth	2	5	10	12	15	16
	Valores obtenidos					
Precisión	0.8626	0.8522	0.8559	0.8518	0.8596	0.8559
Tasa de fallo	0.1374	0.1478	0.1441	0.1482	0.1404	0.1441
Sensibilidad	0.8626	0.8522	0.8559	0.8518	0.8596	0.8559
Especificidad	0.9700	0.9680	0.9688	0.9680	0.9698	0.9692
PV+	0.8709	0.8619	0.8663	0.8622	0.8683	0.8629
PV-	0.9718	0.9696	0.9705	0.9698	0.9714	0.9705
F1-score	0.8590	0.8479	0.8504	0.8455	0.8542	0.8504

Tabla 39: Parámetros y métricas de evaluación de los árboles de decisión.

Como se puede apreciar en la tabla, para las diferentes configuraciones probadas con los valores de kNN (2,5,10,12,15,16), se han obtenido los mejores resultados con un NumNeighbors de 2, en el que se ha obtenido una precisión del 86.26 % y una tasa de fallo del 13.74 %. Además, el F1-score obtenido es del 85.9 %.

		Predicción					
		Dama	Peón	Caballo	Alfil	Torre	Rey
Real	Dama	4.50	0.00	0.60	0.10	0.00	0.10
	Peón	0.00	3.70	0.00	0.00	0.00	0.00
	Caballo	0.00	0.00	4.00	0.10	0.10	0.10
	Alfil	0.10	0.00	0.10	4.20	0.00	0.30
	Torre	0.40	0.00	0.10	0.00	3.30	0.20
	Rey	0.70	0.00	0.00	0.60	0.10	3.40

Tabla 40: Mejor matriz de confusión para kNN.

4.5.3. Discusión

Tras la ejecución de esta aproximación, con la adición de la figura del rey, hemos observado una disminución del rendimiento de las diversas arquitecturas. Para las ANN, se obtuvo una precisión del 79.61 %, teniendo esta una topología de 16 neuronas, siendo esta la peor técnica. Para los árboles de decisión, se obtuvo una precisión del 87.79 %, teniendo 13 y 14 como profundidad máxima, siendo esta la técnica con la precisión más alta. En los k-vecinos, se obtuvo una precisión del 86.26 %, con un número de vecinos de 2. Por otra parte, en los SVM, se obtiene una precisión del 85.49 %, usando el kernel rbf y un valor de C de 3.

Debido a los resultados obtenidos, en la siguiente aproximación añadiremos una nueva serie de características de tal forma que los modelos que no diferencian correctamente las piezas de la torre, el caballo y el rey del resto de piezas, lo hagan correctamente.

4.6. Sexta aproximación

4.6.1. Descripción

Después de comprobar los resultados de la anterior aproximación nos dimos cuenta de que la mayoría de los problemas estaban en la capacidad de los modelos de diferenciar la torre, el caballo y el rey de las demás piezas. Para resolver este problema decidimos añadir una nueva característica para cada una de estas piezas.

Para el caballo añadimos una característica que identifica la simetría respecto al eje Y, ya que esta es la única pieza que no es simétrica.

Para obtener este valor se ha calculado la boundig box y se ha pasado la imagen dentro de esta bounding box a píxeles negros y blancos, dividimos esta imagen a la mitad respecto al eje Y y comparamos en cada píxel y su simétrico en la otra imagen si ambos son blancos o ambos son negros. Luego dividimos

el número de píxeles que son iguales entre el total de píxeles, este será el valor que usaremos para una de las nuevas características.

Figura 5: Simetría Y caballo, 1

Figura 6: Simetría Y caballo, 2

Figura 7: Simetría Y torre

En la última imagen los píxeles negros representan una zona que no es simétrica

Para la torre utilizamos la simetría respecto al eje X del tercio medio de la imagen.

Para obtener este valor se utilizo el mismo sistema que para la anterior característica pero respecto al eje X y solo del tercio medio de la imagen.

Figura 8: Simetría X torre, 1

Figura 9: Simetría X torre, 2

Figura 10: Simetría X dama

Al igual que en el anterior ejemplo en la última imagen los píxeles negros representan una zona que no es simétrica

Para el rey utilizamos la distancia entre el primer y último píxel de la línea horizontal media de la imagen ya que el rey es la pieza más ancha en el medio.

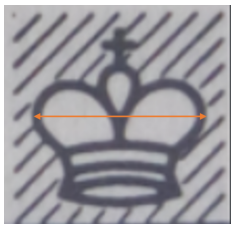

Figura 11: Ancho rey en el medio

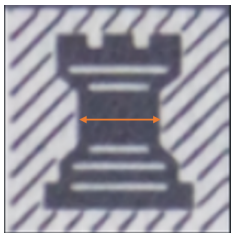

Figura 12: Ancho torre en el medio

4.6.2. Resultados

En este apartado explicaremos las distintas características que empleamos con cada técnica y los resultados obtenidos tras sus ejecuciones. Estos resultados

son el promedio de los tests de los 10 folds.

Resultados de Redes Neuronales								
	Configuraciones probadas							
Topología	[2]	[8]	[16]	[4, 2]	[8, 4]	[8, 4, 2]	[2, 8]	[16, 8]
	Valores obtenidos							
Precisión	0.7874	0.9043	0.9058	0.6489	0.8556	0.5583	0.8027	0.8882
Tasa de fallo	0.2126	0.0957	0.0942	0.3511	0.1444	0.4417	0.1973	0.1118
Sensibilidad	0.7874	0.9043	0.9058	0.6489	0.8556	0.5583	0.8027	0.8882
Especificidad	0.9540	0.9804	0.9805	0.9247	0.9699	0.9023	0.9563	0.9767
PV+	0.7822	0.9207	0.9218	0.5677	0.8624	0.4195	0.8001	0.9054
PV-	0.9606	0.9804	0.9806	0.9388	0.9717	0.9256	0.9625	0.9767
F1-score	0.7645	0.9039	0.9054	0.5787	0.8472	0.4567	0.7847	0.8887

Tabla 41: Parámetros y métricas de evaluación de las redes neuronales.

Como se puede apreciar en la tabla anterior, para las redes neuronales se han obtenido los mejores resultados en la configuración que tiene una topología de 16 neuronas. Para esta, la precisión obtenida es del 90.58 %, siendo la más alta de todas las configuraciones probadas. Además, también presenta una tasa de fallo del 9.42 % y un F1-score del 90.54 %. Interesante mencionar que con una topología de [8] el valor se acerca mucho al mejor resultado obtenido, habiendo solo una diferencia de 0.15 % entre sus valores de precisión,de tasas de fallo y F1-score.

Los siguientes parámetros se han utilizado con todas las topologías de redes neuronales:

- LearningRate = 0.01
- ValidationRatio = 0
- NumRepetitionsANNTraining = 10
- NumMaxEpochs = 400
- MaxEpochsVal = 6

		Predicción					
		Dama	Peón	Caballo	Alfil	Torre	Rey
Real	Dama	4.62	0.00	0.50	0.00	0.03	0.15
	Peón	0.00	3.63	0.03	0.00	0.04	0.00
	Caballo	0.04	0.00	4.13	0.00	0.00	0.13
	Alfil	0.03	0.00	0.03	4.41	0.09	0.14
	Torre	0.21	0.13	0.12	0.00	3.54	0.00
	Rey	0.43	0.00	0.32	0.10	0.00	3.95

Tabla 42: Mejor matriz de confusión para redes neuronales.

Resultados de SVM								
	Configuraciones probadas							
Kernel	rbf	linear	poly	sigmoid	rbf	linear	poly	sigmoid
C	1	1	1	1	3	3	3	3
	Valores obtenidos							
Precisión	0.9178	0.9181	0.9365	0.1893	0.9178	0.9141	0.9402	0.0925
Tasa de fallo	0.0822	0.0819	0.0635	0.8107	0.0822	0.0859	0.0598	0.9075
Sensibilidad	0.9178	0.9181	0.9365	0.1893	0.9178	0.9141	0.9402	0.0925
Especificidad	0.9832	0.9827	0.9869	0.8042	0.9826	0.9825	0.9878	0.7878
PV+	0.9371	0.9386	0.9477	0.0871	0.9329	0.9361	0.9508	0.0700
PV-	0.9829	0.9832	0.988	0.8365	0.9834	0.9821	0.9884	0.7911
F1-score	0.9183	0.9187	0.9338	0.0908	0.9171	0.9152	0.9391	0.0663

Tabla 43: Parámetros y métricas de evaluación de SVM.

Los siguientes parámetros se han utilizado con todas las configuraciones de SVM:

- KernelDegree = 3
- KernelGamma = 2

Como se puede apreciar en la tabla, para las distintas configuraciones probadas con los diferentes tipos de kernel (rbf,linear,poly,sigmoid), con valores de C de 1 y 3 para cada uno de ellos, se han obtenido los mejores resultados con el kernel poly con C 3. Se ha obtenido una precisión del 94.02 % y una tasa de fallo del 05.98 %. Además, el F1-score obtenido es del 93.91 %.

		Predicción					
		Dama	Peón	Caballo	Alfil	Torre	Rey
Real	Dama	4.90	0.00	0.40	0.00	0.00	0.00
	Peón	0.00	3.70	0.00	0.00	0.00	0.00
	Caballo	0.20	0.00	4.00	0.10	0.00	0.00
	Alfil	0.00	0.00	0.00	4.60	0.00	0.10
	Torre	0.20	0.20	0.00	0.10	3.40	0.10
	Rey	0.00	0.00	0.10	0.10	0.00	4.60

Tabla 44: Mejor matriz de confusión para SVM.

Resultados de Árboles de Decisión						
	Configuraciones probadas					
MaxDepth	2	4	8	10	13	14
	Valores obtenidos					
Precisión	0.5185	0.8190	0.9293	0.9331	0.9331	0.9331
Tasa de fallo	0.4815	0.1810	0.0707	0.0669	0.0669	0.0669
Sensibilidad	0.5185	0.8190	0.9293	0.9331	0.9331	0.9331
Especificidad	0.8822	0.9673	0.9848	0.9858	0.9858	0.9858
PV+	0.4105	0.7991	0.9422	0.9445	0.9445	0.9445
PV-	0.9253	0.9692	0.9862	0.9869	0.9869	0.9869
F1-score	0.4239	0.7925	0.9282	0.9318	0.9318	0.9318

Tabla 45: Parámetros y métricas de evaluación de los árboles de decisión.

La tabla anterior presenta los resultados de evaluación de diferentes configuraciones de árboles de decisión, donde se varió el parámetro de MaxDepth. Se puede observar que para la profundidad máxima de 10, 13 y 14 se ha obtenido la precisión más alta, siendo esta del 93.31 %. Además, se ha obtenido para esta una tasa de fallo del 6.69 % y un F1-score del 93.18 %.

		Predicción					
		Dama	Peón	Caballo	Alfil	Torre	Rey
Real	Dama	5.20	0.00	0.00	0.00	0.10	0.00
	Peón	0.00	3.60	0.10	0.00	0.00	0.00
	Caballo	0.20	0.00	3.80	0.20	0.10	0.00
	Alfil	0.00	0.00	0.10	4.30	0.00	0.30
	Torre	0.20	0.00	0.00	0.00	3.70	0.10
	Rey	0.00	0.00	0.00	0.10	0.30	4.40

Tabla 46: Mejor matriz de confusión para árboles de decisión.

Resultados de kNN						
	Configuraciones probadas					
MaxDepth	2	5	10	12	15	16
	Valores obtenidos					
Precisión	0.9397	0.9101	0.9029	0.9029	0.9029	0.9029
Tasa de fallo	0.0603	0.0899	0.0971	0.0971	0.0971	0.0971
Sensibilidad	0.9397	0.9101	0.9029	0.9029	0.9029	0.9029
Especificidad	0.9868	0.9809	0.9795	0.9795	0.9795	0.9795
PV+	0.9525	0.9266	0.9226	0.9226	0.9236	0.9236
PV-	0.988	0.9814	0.9797	0.9797	0.9797	0.9797
F1-score	0.9392	0.9098	0.9033	0.9033	0.9036	0.9036

Tabla 47: Parámetros y métricas de evaluación de los árboles de decisión.

Como se puede apreciar en la tabla, para las diferentes configuraciones probadas con los valores de kNN (2,5,10,12,15,16), se han obtenido los mejores resultados con un NumNeighbors de 2, en el que se ha obtenido una precisión del 93.97 % y una tasa de fallo del 06.03 %. Además, el F1-score obtenido es del 93.92 %.

		Predicción					
		Dama	Peón	Caballo	Alfil	Torre	Rey
Real	Dama	5.10	0.00	0.20	0.00	0.00	0.00
	Peón	0.00	3.70	0.00	0.00	0.00	0.00
	Caballo	0.00	0.00	4.20	0.10	0.00	0.00
	Alfil	0.20	0.00	0.00	4.50	0.00	0.00
	Torre	0.20	0.10	0.10	0.10	3.50	0.00
	Rey	0.30	0.00	0.20	0.10	0.00	4.20

Tabla 48: Mejor matriz de confusión para kNN.

4.6.3. Discusión

Después de añadir estas tres características hemos conseguido que las piezas de rey, torre y caballo sean detectadas con más precisión que en la anterior aproximación. El mayor problema es la identificación de la reina, ya que los distintos modelos aún la confunden con otras piezas. A pesar de ello, la precisión máxima ha subido de la anterior aproximación,desde un 87,79 % en árboles de decisión, a un 94.02 % en SVM. Este porcentaje ya se puede considerar aceptable, y teniendo en cuenta que ya estamos trabajando con todas las piezas, podemos considerar esta aproximación como la final.

4.7. Deep Learning

4.7.1. Descripción

Para esta aproximación utilizamos una nueva técnica de aprendizaje automático: Deep Learning, basada en redes neuronales con capas convolucionales; con el objetivo de comparar una nueva forma de clasificar imágenes con las técnicas estudiadas en anteriores aproximaciones y ver si esta es adecuada para nuestro problema.

Las imágenes utilizadas son las mismas que en la aproximación anterior: 300 imágenes de las 6 piezas del ajedrez (50 de cada una). Para facilitar el uso de Deep Learning, fueron reescaladas a un tamaño fijo de 80x80 píxeles.

Al ser una base de datos pequeña (300 imágenes), influye mucho la partición de los datos en los conjuntos de entrenamiento y test (en este caso no utilizamos conjunto de validación), por lo que una partición aleatoria puede resultar desfavorable y provocar resultados engañosos. Para solucionar esto utilizamos la técnica de validación cruzada con 10 folds, que minimiza estos posibles efectos negativos de una mala división de los datos.

4.7.2. Resultados

Arquitecturas probadas:

- 1: Capa convolucional: filtro 3x3, 1 canal de entrada, 16 de salida
  Capa MaxPool: divide en 2 alto y ancho de las imágenes (80x80 a 40x40)
  Capa convolucional: filtro 3x3, 16 canales de entrada, 32 de salida
  Capa MaxPool: vuelve a divdir en 2 alto y ancho (40x40 a 20x20)
  Capa convolucional: filtro 3x3, 32 canales de entrada, 32 de salida
  Capa MaxPool: vuelve a divdir en 2 alto y ancho (20x20 a 10x10)
  Reshape: convierte las imágenes de matrices 3D a arrays
  Capa Dense: 3200 entradas, 6 salidas
  Capa softmax

- 2: Capa convolucional: filtro 3x3, 1 canal de entrada, 16 de salida
  Capa MaxPool: divide en 2 alto y ancho de las imágenes (80x80 a 40x40)
  Capa convolucional: filtro 3x3, 16 canales de entrada, 32 de salida
  Capa MaxPool: vuelve a divdir en 2 alto y ancho (40x40 a 20x20)
  Reshape: convierte las imágenes de matrices 3D a arrays
  Capa Dense: 12800 entradas, 6 salidas
  Capa softmax

- 3: Capa convolucional: filtro 3x3, 1 canal de entrada, 16 de salida
  Capa MaxPool: divide en 2 alto y ancho de las imágenes (80x80 a 40x40)
  Reshape: convierte las imágenes de matrices 3D a arrays
  Capa Dense: 25600 entradas, 6 salidas
  Capa softmax

- 4: Capa convolucional: filtro 3x3, 1 canal de entrada, 8 de salida
  Capa MaxPool: divide en 2 alto y ancho de las imágenes (80x80 a 40x40)
  Capa convolucional: filtro 3x3, 8 canales de entrada, 16 de salida
  Capa MaxPool: vuelve a divdir en 2 alto y ancho (40x40 a 20x20)
  Capa convolucional: filtro 3x3, 16 canales de entrada, 32 de salida
  Capa MaxPool: vuelve a divdir en 2 alto y ancho (20x20 a 10x10)
  Reshape: convierte las imágenes de matrices 3D a arrays
  Capa Dense: 3200 entradas, 6 salidas
  Capa softmax

- 5: Capa convolucional: filtro 3x3, 1 canal de entrada, 8 de salida
  Capa MaxPool: divide en 2 alto y ancho de las imágenes (80x80 a 40x40)
  Capa convolucional: filtro 3x3, 8 canales de entrada, 16 de salida
  Capa MaxPool: vuelve a divdir en 2 alto y ancho (40x40 a 20x20)
  Reshape: convierte las imágenes de matrices 3D a arrays
  Capa Dense: 6400 entradas, 6 salidas
  Capa softmax

- 6: Capa convolucional: filtro 3x3, 1 canal de entrada, 8 de salida
  Capa MaxPool: divide en 2 alto y ancho de las imágenes (80x80 a 40x40)

Reshape: convierte las imágenes de matrices 3D a arrays
Capa Dense: 12800 entradas, 6 salidas
Capa softmax

- 7: Capa convolucional: filtro 2x2, 1 canal de entrada, 16 de salida
  Capa MaxPool: divide en 2 alto y ancho de las imágenes (80x80 a 40x40)
  Capa convolucional: filtro 3x3, 16 canales de entrada, 32 de salida
  Capa MaxPool: vuelve a divdir en 2 alto y ancho (40x40 a 20x20)
  Reshape: convierte las imágenes de matrices 3D a arrays
  Capa Dense: 12800 entradas, 6 salidas
  Capa softmax

- 8: Capa convolucional: filtro 3x3, 1 canal de entrada, 8 de salida
  Capa MaxPool: divide en 2 alto y ancho de las imágenes (80x80 a 40x40)
  Capa convolucional: filtro 3x3, 8 canales de entrada, 16 de salida
  Capa MaxPool: vuelve a divdir en 2 alto y ancho (40x40 a 20x20)
  Reshape: convierte las imágenes de matrices 3D a arrays
  Capa Dense: 6400 entradas, 6 salidas
  Capa softmax

Resultados de Deep Learning								
Arquitectura	1	2	3	4	5	6	7	8
Precisión	0.9811	0.9956	0.9922	0.9822	0.9889	0.9867	0.9878	0.9767
Tasa de fallo	0.0189	0.0044	0.0078	0.0178	0.0111	0.0133	0.0122	0.0233
Sensibilidad	0.9433	0.9867	0.9767	0.9467	0.9667	0.9600	0.9633	0.9300
Especificidad	0.9887	0.9973	0.9953	0.9893	0.9933	0.9920	0.9927	0.9860
PV+	0.9564	0.9889	0.9800	0.9587	0.9746	0.9679	0.9718	0.9469
PV-	0.9896	0.9974	0.9955	0.9903	0.9937	0.9923	0.9932	0.9871
F1-score	0.9385	0.9865	0.9760	0.9405	0.9657	0.9597	0.9607	0.9250

Tabla 49: Parámetros y métricas de evaluación de Deep Learning.

4.7.3. Discusión

Tras la ejecución de esta aproximación, la precisión más alta obtenida es de un 99.56 %, la cual se considera un resultado bastante satisfactorio. Esta precisión no se encuentra ni en la red neuronal más compleja, ni en la más sencilla, por lo que creemos que se encontró una arquitectura adecuada para nuestro proyecto, aunque esto puede deberse también a la aleatoriedad del proceso, ya que otras arquitecturas dan resultados similares.

Además, en comparación con la precisión máxima obtenida en la sexta aproximación (94.02 %), se puede observar que se mejora con todas las arquitecturas probadas usando esta técnica. Esto nos da a entender que el uso de deep learning es adecuado para el análisis de imágenes, ya que sin la extracción de características, es capaz de obtener un resultado mejor que las técnicas usadas en la sexta aproximación.

5. Conclusión

Las conclusiones que se pueden sacar son limitadas, ya que de momento el trabajo no ha sido terminado. Aunque, se puede apreciar que los resultados mejoran considerablemente por iteración. El mejor sistema en hasta este momento es kNN, en cuanto a los peores sistemas, varía según la iteración pero son las ANN y SVM. Consideramos que es viable en general el uso de estos sistemas debido al promedio de resultados obtenidos.

Algunas de las principales ventajas que se pueden obtener mediante el uso de este sistema son: su sencillez y claridad, pequeño tamaño de la base de datos y poca necesidad computacional. El sistema está formado por arquitecturas muy sencillas y eficientes, siempre tratando de obtener los mejores valores en tiempos óptimos. Lo segundo, el sistema no necesita una gran cantidad de fotos para su correcto funcionamiento. Por último, su poca necesidad computacional. El sistema tarda muy poco en entrenarse, y no se necesita una máquina muy potente para ello, lo cual lo hace adecuado para muchos tareas.

Durante el desarrollo de la memoria, uno de los obstáculos más importantes que encontramos fue el uso de LaTeX, debido a la inclusión de numerosas tablas. En relación a la implementación de la práctica, el mayor desafío al que nos enfrentamos fue el entrenamiento de las redes neuronales artificiales, ya que no teníamos mucha experiencia previa en este ámbito.

6. Trabajo futuro

A partir de los resultados obtenidos en este trabajo, se observa que una aplicación de este estilo podría ser utilizada en muchos campos. Aunque las reglas de extracción de características sean específicas del ajedrez, si estas se modifican, estas técnicas se podrían aplicar a multitud de situaciones, ya sean otros juegos de mesa, como las damas, o el Go, u otros problemas de clasificación de imágenes simples.

Si se quisiera mejorar este sistema para que fuera aplicado a otros casos, en general algunos más complejos, se necesitaría profundizar en técnicas de procesamiento de imágenes para la extracción de características, ya que actualmente

está bastante limitado. Otro aspecto a considerar sería la calidad y cantidad de las imágenes de la base de datos, ya que podría ser corta para otros escenarios.

7. Referencias

Origen de la base de datos

[1] Moskalenko, V. (2010). Revoluciona tu ajedrez III: Aperturas. Barcelona: Ediciones Robinbook.

[2] Alekhine, A. (1924-1937). Mis mejores partidas. Madrid: Editorial La Casa del Ajedrez.

[3] Krammik, V. (2015). Partidas magistrales volumen I. Asturias: Editorial Chessy.

[4] Smymslov, V. (2006). El virtuoso de los finales. Barcelona: Editorial Hispano Europea.

Trabajos destacados en el tema

[1] Laskowski, A. (1986). Chessboard and chess piece recognition with the support of neural networks. In *International Joint Conference on Neural Networks* (pp. 165-170). IEEE.

[2] del Barrio García, A. A. (2020). Técnicas de aceleración para el reconocimiento de piezas de ajedrez (Tesis de maestría). Universidad Complutense de Madrid.

[3] Paniagua Benito, M.(2017). Trabajo de Fin de Máster: La influencia del ajedrez en los procesos cognitivos.Universidad Internacional de la Rioja, Facultad de Educación.

[4] Molina Nuñez,A. , Plaza García-Abadillo, C. , Rengel Lazcano, K. (2021-2022) Software de reconocimiento de piezas de ajedrez en tiempo real. Universidad Complutense de Madrid, Facultad de informática.

[5] Sanango Peña, J. E. (2019). Trabajo de Fin de Máster: Estudio y desarrollo de los algoritmos de visión y de Inteligencia Artificial aplicados a un robot, para resolver partidas de ajedrez hombre-máquina. Terrassa, España: Escuela Superior de Ingenierías Industrial, Aeroespacial y Audiovisual de Terrassa.

[6] Akinola, A. M., and Adeyemo, T. A. (2019). Deep learning based chess piece recognition and tracking for chess game analysis. In *2019 International*

Conference on Computing, Networking and Communications (ICNC) (pp. 166-170). IEEE.

[7] Hamdan, M. H., and Lu, S. (2020). An efficient and real-time chess piece recognition system based on deep learning. In *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)* (pp. 1-6). IEEE.

[8] Saad, W., and Rebai, A. (2020). A hybrid deep learning approach for chess piece recognition. *Journal of Ambient Intelligence and Humanized Computing*, 11(6), 2183-2195.

[9] Ding, J., Liu, J., Li, H., and Li, H. (2019). Automatic chess piece recognition with enhanced YOLOv3. In *IEEE International Conference on Industrial Technology* (pp. 1968-1973). IEEE.