Instituto Politécnico do Porto

Licenciatura em Engenharia Informática

Ano letivo 2024/2025

Unidade Curricular de Inteligência Artificial

Trabalho Prático - Chess AI Assistant

Grupo 14:

António Silva – 8220207

Guilherme Barreiro – 8220849

Tiago Pacheco – 8220208

Docente:

Davide Carneiro

Índice

[1. Introdução 3](#_Toc196661192)

[2. Definição do Problema 3](#_Toc196661193)

[Contextualização da Tarefa 3](#_Toc196661194)

[Relevância da Tarefa 3](#_Toc196661195)

[3. Recolha de Imagens 4](#_Toc196661196)

[Geração Automática de Tabuleiros Aleatórios 4](#_Toc196661197)

[Justificação da Abordagem 5](#_Toc196661198)

[Upload e Preparação para Etiquetagem 5](#_Toc196661199)

[4. Etiquetagem dos Dados 6](#_Toc196661200)

[5. Técnicas de Data Augmentation 6](#_Toc196661201)

[6. Treino dos Modelos 7](#_Toc196661202)

[7. Avaliação dos Modelos 8](#_Toc196661203)

[Resultados Observados 8](#_Toc196661204)

[Modelo ChessBoardDetection 5: 8](#_Toc196661205)

[Modelo ChessBoardDetection 2: 8](#_Toc196661206)

[Análise Crítica 8](#_Toc196661207)

[Imagens de Avaliação 9](#_Toc196661208)

[8. Implementação em Produção 11](#_Toc196661209)

[9. Discussão dos Resultados e Melhorias Futuras 11](#_Toc196661210)

[10. Conclusão 11](#_Toc196661211)

[11. Referências 11](#_Toc196661212)

[12. Anexos (ex. prints, gráficos, resultados) 11](#_Toc196661213)

a

# 1. Introdução

O presente relatório descreve o desenvolvimento de um projeto prático no âmbito da unidade curricular de Inteligência Artificial. Este projeto tem como principal objetivo aplicar técnicas de Machine Learning, em particular através da arquitetura YOLOv8, para automatizar tarefas relevantes num domínio específico, neste caso, xadrez.

O xadrez, sendo um jogo com regras bem definidas e uma estrutura visual consistente, apresenta um cenário ideal para a aplicação de técnicas de deteção de objetos.

Neste projeto, foi desenvolvida uma aplicação que utiliza imagens capturadas do tabuleiro para identificar as peças e a sua posição, convertendo essa informação para a notação FEN (Forsyth–Edwards Notation), muito utilizada em engines e plataformas de xadrez.

Este trabalho envolveu a recolha e anotação de dados, passando pelo treino e avaliação de modelos, até à implementação de uma solução funcional em produção, capaz de analisar o estado do jogo em tempo real. O projeto não só reforça os conhecimentos adquiridos ao longo da UC, como demonstra a aplicabilidade prática da IA em contextos estratégicos e educacionais.

# 2. Definição do Problema

O problema abordado neste projeto insere-se na área da Visão Computacional aplicada a jogos de tabuleiro. O objetivo principal é desenvolver um sistema capaz de reconhecer automaticamente a posição das peças num tabuleiro de xadrez a partir de uma imagem, convertendo essa informação para a notação padrão FEN (Forsyth–Edwards Notation).

## Contextualização da Tarefa

No contexto do projeto, a tarefa automatizada foi definida como:

Detetar, classificar e localizar as peças de xadrez num tabuleiro, a partir de imagens capturadas.

Isto implica resolver três sub-problemas fundamentais:

* Identificar corretamente todas as peças visíveis (peão, cavalo, bispo, torre, dama e rei) e distinguir as suas cores (branco ou preto).
* Determinar a posição exata de cada peça no tabuleiro (casa de xadrez correspondente, ex.: e4, d5, etc.).
* Gerar uma representação FEN precisa a partir das peças detetadas.

## Relevância da Tarefa

Automatizar a leitura do estado de um tabuleiro de xadrez tem grande valor no contexto da modernização de práticas associadas ao jogo, nomeadamente:

* Facilitar o acompanhamento e análise de partidas físicas.
* Serve como base para sistemas de recomendação de jogadas e análises de estratégias.

Esta tarefa é particularmente desafiante, pois exige:

* Um alto grau de precisão na deteção de objetos pequenos e semelhantes entre si (peças pretas e brancas).
* Capacidade de generalização a diferentes configurações de tabuleiros e peças.

Caracterização do Processo de Integração

A integração do sistema final com a aplicação consistirá na captura do ecrã (ou imagem estática) do tabuleiro de xadrez. Esta imagem será passada ao modelo YOLO treinado, que devolverá as localizações das peças detetadas.

Com esta informação, o sistema reconstruirá automaticamente a posição atual do tabuleiro em formato FEN, sendo posteriormente ser usada para recomendações de jogadas e interação com o melhor engine atual, Stockfish 17.

# 3. Recolha de Imagens

A fase de recolha de imagens representou um passo fundamental no desenvolvimento do nosso modelo de visão computacional para reconhecimento de peças de xadrez. Como a tarefa do projeto exige que a inteligência artificial reconheça corretamente diferentes tipos de peças (peões, cavalos, torres, etc.) em posições variadas sobre um tabuleiro, era essencial reunir um dataset diversificado, balanceado e de qualidade.

## Geração Automática de Tabuleiros Aleatórios

Para garantir essa diversidade, desenvolvemos um script em Python “scriptImages.py” que recorre à biblioteca python-chess para gerar tabuleiros de xadrez com peças dispostas de forma aleatória. A função random\_board() foi responsável por criar cenários únicos, em que cada imagem contém entre 5 a 20 peças posicionadas aleatoriamente, com diferentes combinações de peças brancas e pretas.

As imagens foram inicialmente geradas em formato SVG com chess.svg, sendo posteriormente convertidas para formato PNG com cairosvg, permitindo a sua utilização posterior em ferramentas de anotação.

Para assegurar a organização dos dados e permitir a criação de múltiplos datasets, as imagens foram divididas em duas pastas distintas: images0-100/ e images100-200/, cada uma contendo 100 imagens. O dataset "ChessBoardDetection 2" foi criado a partir das imagens presentes em images0-100/, enquanto o dataset "ChessBoardDetection 5" utilizou as imagens da pasta images100-200/.

Uma imagem com jogo de tabuleiro, Jogos, Jogos e desportos indoor, jogador de xadrez

Os conteúdos gerados por IA poderão estar incorretos.

Exemplo de tabuleiro gerado automaticamente “board\_36.png”

## Justificação da Abordagem

Esta abordagem permitiu-nos simular centenas de posições únicas, representando diferentes cenários possíveis de uma partida de xadrez real. Além disso, ao controlar totalmente a geração das imagens, conseguimos evitar a introdução de ruído visual desnecessário e garantir que as imagens estão alinhadas com a resolução recomendada (640x640) para o treino dos modelos YOLO.

## Upload e Preparação para Etiquetagem

Após a geração das imagens, procedemos ao seu upload para a plataforma Roboflow, tal como sugerido no enunciado do projeto. Esta plataforma permitiu-nos não só fazer a anotação manual dos objetos (peças de xadrez), como também dividir o dataset em conjuntos de treino, validação e teste.

Esta metodologia assegurou que o modelo seria treinado com uma variedade suficientemente grande de exemplos, maximizando assim a sua capacidade de generalização a novas imagens captadas durante a execução do projeto em tempo real.

# 4. Etiquetagem dos Dados

Após a geração dos datasets de imagens de tabuleiros de xadrez, foi necessário realizar a etiquetagem dos dados, de forma a preparar o conjunto de treino para o modelo de deteção de objetos.

A etiquetagem foi feita utilizando a plataforma Roboflow, através da criação de dois projetos:

* ChessBoardDetection 2
* ChessBoardDetection 5

Em cada projeto, realizámos:

* Upload das imagens geradas.
* Criação de bounding boxes para cada peça de xadrez, anotando o tipo de peça e a respetiva cor (branco ou preto).
* Divisão automática do dataset em treino, validação e teste.

As classes usadas foram: Peão, Cavalo, Bispo, Torre, Dama e Rei, para ambos os lados (branco e preto), totalizando 12 classes no total.

Este processo assegurou que cada imagem estava corretamente anotada, fornecendo ao modelo dados de alta qualidade para treino e maximizando a capacidade de generalização.

# 5. Técnicas de Data Augmentation

Com o objetivo de aumentar a robustez do modelo e melhorar a sua capacidade de generalização, aplicámos uma técnica específica de data augmentation: a aplicação de efeito blur nas imagens.

O efeito blur (suavização da imagem) foi utilizado para introduzir variações subtis na nitidez das imagens, simulando possíveis imperfeições ou capturas com baixa qualidade que poderiam ocorrer em cenários reais. Esta abordagem força o modelo a aprender a reconhecer as peças mesmo em condições visuais menos ideais.

A técnica de data augmentation foi aplicada apenas ao dataset utilizado para o treino do modelo ChessBoardDetection 5. O dataset associado ao modelo ChessBoardDetection 2 não passou por qualquer processo de aumento de dados, permitindo posteriormente comparar diretamente o impacto da utilização do blur na performance dos modelos.

A aplicação de blur revelou-se eficaz, como demonstrado nos resultados de avaliação, onde o modelo ChessBoardDetection 5 apresentou uma capacidade de deteção superior e níveis de confiança mais elevados.

# 6. Treino dos Modelos

Após a criação automática das imagens de tabuleiros de xadrez com posições aleatórias, procedemos ao upload dos ficheiros para a plataforma Roboflow, onde se iniciou o processo de etiquetagem.

Para cada modelo no Roboflow, realizámos a identificação manual de cada peça visível em todas as imagens. Para cada tabuleiro, foram desenhadas bounding boxes à volta de todas as peças, associando-as à respetiva classe (tipo de peça e cor), como demonstrado na imagem abaixo.

Uma imagem com captura de ecrã, quadrado, xadrez

Os conteúdos gerados por IA poderão estar incorretos.

Exemplo da identificação das classes numa imagem

Cada modelo, portanto, foi treinado com base na etiquetagem precisa de 100 imagens únicas, garantindo um conjunto de dados robusto e adequado para a tarefa de deteção.

Este trabalho de anotação foi fundamental para ensinar o modelo a distinguir corretamente entre peças de diferentes tipos e cores, condição necessária para a posterior reconstrução do estado do tabuleiro em notação FEN.

# 7. Avaliação dos Modelos

Após o treino, procedemos à avaliação dos dois modelos desenvolvidos: ChessBoardDetection 2 e ChessBoardDetection 5.

Para a avaliação, utilizámos a mesma imagem de tabuleiro para ambos os modelos, permitindo uma comparação direta da sua capacidade de deteção de peças.

## Resultados Observados

### Modelo ChessBoardDetection 5:

* Conseguiu identificar todas as peças corretamente.
* As percentagens de confiança associadas às deteções foram, de forma geral, elevadas (superiores a 85% na maioria dos casos).
* Nenhuma peça ficou por detetar, e não foram observados falsos positivos.

### Modelo ChessBoardDetection 2:

* Duas peças ficaram por identificar, o que indica falhas na deteção.
* As percentagens de confiança nas detecções foram visivelmente mais baixas em comparação com o ChessBoardDetection 5, com vários valores abaixo de 70%.

## Análise Crítica

Estes resultados sugerem que o modelo ChessBoardDetection 5 é mais robusto e fiável, demonstrando melhor capacidade de generalizar sobre novas imagens de tabuleiros.

Esta superioridade pode ser justificada pela utilização de técnicas de data augmentation, em particular a aplicação de efeito blur, que tornou o modelo mais resistente a pequenas variações visuais nas imagens.

Por outro lado, o modelo ChessBoardDetection 2, treinado com um dataset sem técnicas de aumento de dados, apresentou dificuldades em reconhecer todas as peças, ficando duas peças por identificar e evidenciando uma menor taxa de confiança nas predições.

Além disso, a maior percentagem de confiança associada às deteções feitas pelo modelo ChessBoardDetection 5 mostra que este modelo é mais seguro nas decisões que toma — um fator crítico para aplicações que exigem alta precisão, como a geração automática de posições FEN.

## Imagens de Avaliação

Uma imagem com captura de ecrã, quadrado, Jogos, jogo de tabuleiro

Os conteúdos gerados por IA poderão estar incorretos.

Modelo - ChessBoardDetection 5

Uma imagem com captura de ecrã, quadrado, Jogos, jogo de tabuleiro

Os conteúdos gerados por IA poderão estar incorretos.

Modelo - ChessBoardDetection 2

# 8. Implementação em Produção

Descrição detalhada a ser preenchida com base no projeto desenvolvido.

# 9. Discussão dos Resultados e Melhorias Futuras

Descrição detalhada a ser preenchida com base no projeto desenvolvido.

# 10. Conclusão

Descrição detalhada a ser preenchida com base no projeto desenvolvido.

# 11. Referências

Descrição detalhada a ser preenchida com base no projeto desenvolvido.

# 12. Anexos (ex. prints, gráficos, resultados)

Descrição detalhada a ser preenchida com base no projeto desenvolvido.