

INSTITUTO FEDERAL DE SANTA CATARINA

ANDREY ADRIANO DA ROSA

**Aplicativo de Reconhecimento de Lances de
Xadrez com Visão Computacional**

São José - SC

agosto/2024

APLICATIVO DE RECONHECIMENTO DE LANCES DE XADREZ COM VISÃO COMPUTACIONAL

Projeto de Trabalho de conclusão de curso
apresentado à Coordenadoria do Curso de
Engenharia de Telecomunicações do campus
São José do Instituto Federal de Santa Cata-
rina

Orientador: Prof. Roberto Wanderley da No-
brega, Dr.

Coorientador: Prof. Marcos Moecke, Dr.

São José - SC

agosto/2024

RESUMO

Este trabalho propõe o desenvolvimento de um aplicativo móvel destinado ao reconhecimento de lances de xadrez em partidas físicas utilizando técnicas de visão computacional e aprendizado de máquina. A solução proposta utiliza algoritmos avançados, como redes neurais convolucionais (CNN) e o modelo *You Only Look Once* (YOLO), para mapear o tabuleiro, identificar as peças e reconhecer os movimentos realizados pelos jogadores. A aplicação visa alcançar uma precisão superior a 90%, a depender das condições de iluminação e posicionamento da câmera. Espera-se que esta abordagem ofereça uma alternativa acessível e eficaz aos tabuleiros eletrônicos existentes, facilitando o registro automático de partidas de xadrez.

Palavras-chave: Visão computacional. Xadrez. Reconhecimento por imagem. Redes neurais convolucionais. YOLO.

LISTA DE ILUSTRAÇÕES

Figura 1 – Posição inicial das peças em um jogo de xadrez	9
Figura 2 – Movimentação do bispo	11
Figura 3 – Movimentação da torre	11
Figura 4 – Movimentação da dama	12
Figura 5 – Movimentação do cavalo	12
Figura 6 – Movimentação do peão	13
Figura 7 – Movimentação do rei	13
Figura 8 – Captura <i>en passant</i>	14
Figura 9 – Roque curto para as brancas e roque longo para as pretas	15
Figura 10 – Promoção do peão	15
Figura 11 – Notação do tabuleiro	16
Figura 12 – Desambiguação com cavalo	17
Figura 13 – Desambiguação com peças na mesma coluna	17
Figura 14 – Representação de uma imagem digital bidimensional	18
Figura 15 – Representação de uma imagem digital colorida (modelo RGB)	19
Figura 16 – Exemplo de transformação de perspectiva	19
Figura 17 – Exemplo de aplicação do detector de bordas de Canny	20
Figura 18 – Funcionamento de um neurônio artificial	23
Figura 19 – Uma rede neural profunda	23
Figura 20 – Exemplo de convolução 2D	24
Figura 21 – Exemplo de <i>pooling</i>	25
Figura 22 – Funcionamento do YOLO	26

LISTA DE ABREVIATURAS E SIGLAS

CNN *Convolutional Neural Networks.*

CVAT *Computer Vision Annotation Tool.*

DGT *Digital Game Technology.*

DL *Deep Learning.*

DNN *Deep Neural Networks.*

FIDE *Federação Internacional de Xadrez.*

IA *Inteligência Artificial.*

ML *Machine Learning.*

OpenCV *Open-Source Computer Vision.*

YOLO *You Only Look Once.*

SUMÁRIO

1	INTRODUÇÃO	7
1.1	Objetivos	7
1.1.1	Objetivo geral	7
1.1.2	Objetivos específicos	8
1.2	Justificativa	8
2	FUNDAMENTAÇÃO TEÓRICA	9
2.1	O Xadrez	9
2.1.1	Regras do Xadrez	10
2.1.2	Movimentação das peças	10
2.1.3	Lances especiais	14
2.1.4	Notação Algébrica	16
2.2	Técnicas de Processamento de Imagens	18
2.2.1	Transformação de Perspectiva	19
2.2.2	Detecção de Bordas	20
2.3	Visão Computacional	21
2.3.1	Importância e Aplicações	21
2.3.2	<i>Machine Learning</i>	22
2.3.3	<i>Deep Learning</i>	22
2.3.4	Redes Neurais Convolucionais	24
2.3.4.1	Convolução	24
2.3.4.2	<i>Pooling</i>	24
2.3.4.3	Camada Totalmente Conectada	25
2.3.5	You Only Look Once (YOLO)	25
3	METODOLOGIA	27
3.1	Estudos sobre o Xadrez e Visão Computacional	27
3.2	Coleta e Preparação de Imagens	27
3.3	Testes do modelo treinado	27
3.4	Desenvolvimento e ferramentas	28
4	PROPOSTA	29
4.1	Preparação do Conjunto de Dados e Treinamento do Modelo	29
4.2	Descrição do Funcionamento da Aplicação	29
4.2.1	Detecção e Mapeamento do Tabuleiro	29
4.2.2	Reconhecimento das Peças Presentes	30

4.2.3	Reconhecimento do Lance Realizado	30
4.2.4	Geração da Notação Algébrica de Xadrez	30
4.3	Cronograma de Atividades	30
	REFERÊNCIAS	31

1 INTRODUÇÃO

Nos últimos anos, a [Inteligência Artificial \(IA\)](#) tem se evidenciado cada vez mais com o surgimento de novas tecnologias. A visão computacional, um dos ramos da IA, engloba o processamento de imagens e reconhecimento de padrões e proporcionou diversas inovações no ramo da tecnologia, como veículos autônomos, análise de imagens médicas e aplicações em segurança e monitoramento ([FORBES, 2023](#)).

O xadrez é um dos jogos de tabuleiro mais antigos da humanidade. Embora haja controvérsias quanto a sua origem, os primeiros registros do jogo que deu origem ao xadrez, chamado *chaturanga*, datam do século VI, na Índia. Posteriormente, o jogo se espalhou pelo mundo e sofreu alterações até chegar à Europa, onde se estabeleceram as regras modernas entre os séculos XV e XVI ([MURRAY, 2015](#)).

Atualmente, nos tempos da *world wide web*, o xadrez pode ser jogado *online*, permitindo que pessoas de todo o mundo se enfrentem em partidas de diversas modalidades, através de plataformas como a *Chess.com*¹. Apesar disso, torneios presenciais ainda são muito comuns, tanto de grande porte como o Campeonato Mundial de Xadrez, administrado pela [Federação Internacional de Xadrez \(FIDE\)](#)², quanto torneios de pequeno porte organizados por clubes locais.

Os estudos da inteligência artificial voltados ao xadrez são quase tão antigos quanto a própria ciência da computação, iniciando-se na década de 1950, com Alan Turing e Claude Shannon, que criaram os primeiros algoritmos capazes de jogar xadrez. Em 1957, Alex Bernstein, engenheiro da IBM, criou a primeira máquina completamente automatizada capaz de jogar xadrez ([HEATH; ALLUM, 1997](#)).

1.1 Objetivos

1.1.1 Objetivo geral

Desenvolver um aplicativo capaz de fazer o reconhecimento por imagem de lances em um jogo de xadrez utilizando algoritmos de visão computacional e aprendizado de máquina e os apresentar ao usuário em notação algébrica de xadrez.

¹ <https://www.chess.com/>

² <https://www.fide.com/>

1.1.2 Objetivos específicos

- Investigar e aplicar técnicas e algoritmos de visão computacional para detecção de objetos;
- Detectar isoladamente o tabuleiro de xadrez e mapear suas casas, reconhecendo individualmente cada peça presente no tabuleiro;
- Reconhecer todos os lances de um jogo de xadrez utilizando a combinação de visão computacional e aprendizado de máquina;
- Realizar a captura das imagens através de um aplicativo para *smartphones* e mostrar ao usuário cada lance em notação algébrica de xadrez.

1.2 Justificativa

Existem poucas alternativas acessíveis para determinar a posição das peças em tabuleiros físicos de xadrez ou registrar partidas de forma automática. Atualmente, isso é feito manualmente ou utilizando tabuleiros eletrônicos, como o da marca *Digital Game Technology (DGT)*. Este último é usado em torneios de xadrez, porém seu preço ainda é demasiadamente elevado. A marca *DGT*, por exemplo, possui apenas uma loja representante oficial no Brasil³, cujo tabuleiro eletrônico mais barato custa quase R\$4.000,00. Desta forma, o presente trabalho visa proporcionar uma alternativa mais acessível para o registro automático de partidas de xadrez.







³ <https://www.pontodoxadrez.com.br/>

2 FUNDAMENTAÇÃO TEÓRICA

2.1 O Xadrez

O xadrez é um jogo em que dois oponentes movem suas peças em um tabuleiro de 8 fileiras e 8 colunas, com o objetivo de pôr o rei oponente em uma posição que não exista qualquer movimento legal a ser feito, a posição de xeque-mate (FIDE, 2022).

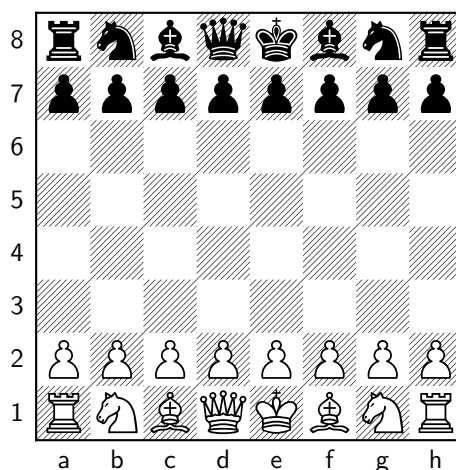
Quadro 1 – Peças do xadrez

Nome (<i>inglês</i>)	Letra Representativa	Símbolo Representativo
Rei (<i>King</i>)	K	
Dama (<i>Queen</i>)	Q	
Bispo (<i>Bishop</i>)	B	
Cavalo (<i>Knight</i>)	N	
Torre (<i>Rook</i>)	R	
Peão (<i>Pawn</i>)	P	

Fonte: Elaborada pelo autor.

Inicialmente, 32 peças são dispostas no tabuleiro, sendo 1 rei, 1 dama, 2 bispos, 2 cavalos, 2 torres e 8 peões de cada cor, cujos símbolos podem ser observados no [Quadro 1](#). A posição inicial das peças no tabuleiro pode ser observada na [Figura 1](#). As casas do tabuleiro são designadas por letras nas colunas e número nas fileiras, assim cada casa pode ser designada por um par letra - número.

Figura 1 – Posição inicial das peças em um jogo de xadrez



Fonte: Elaborada pelo autor.

2.1.1 Regras do Xadrez

As regras oficiais do xadrez são definidas pela [FIDE](#) e conhecidas como as Leis do Xadrez, descritas em um documento disponível publicamente no site oficial da federação ([FIDE, 2022](#)).

O jogador que controla as peças brancas sempre joga primeiro, em seguida é a vez daquele que controla as peças pretas. Eles seguem alternando a vez até o fim do jogo. Um jogador tem a vez apenas quando o oponente tiver finalizado seu lance. O jogador cujo rei sofrer um xeque-mate terá perdido a partida.

Se ocorrer uma situação em que nenhum dos dois jogadores possam realizar um xeque-mate, chamada de “posição morta”, o jogo terá empatado. Também será considerado empate se o jogador que tiver a vez não puder realizar qualquer movimento legal e seu rei não estiver em xeque, neste caso diz-se que a partida terminou com o rei “afogado”. A partida também pode terminar empatada mediante comum acordo entre os dois jogadores, contanto que ambos tenham feito ao menos o primeiro lance.

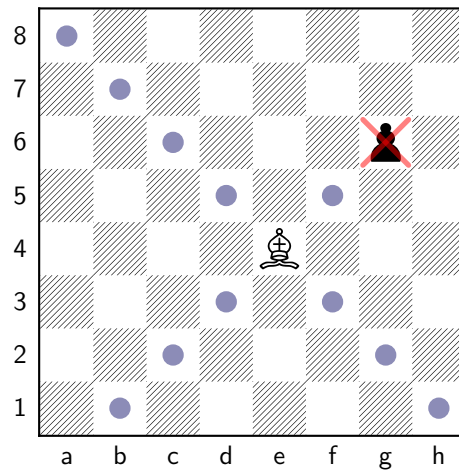
2.1.2 Movimentação das peças

Cada peça do xadrez segue diferentes regras de movimentação no tabuleiro, com suas peculiaridades em relação à direção e quantas casas podem andar. Não é permitido mover uma peça para uma casa em que há uma peça de mesma cor. Se uma peça move-se para uma casa em que há uma peça da outra cor, é realizada uma captura ([FIDE, 2022](#)).

Bispo

O bispo move-se apenas na diagonal, quantas casas forem possíveis, e captura peças oponentes seguindo a mesma regra, conforme demonstrado na [Figura 2](#).

Figura 2 – Movimentação do bispo

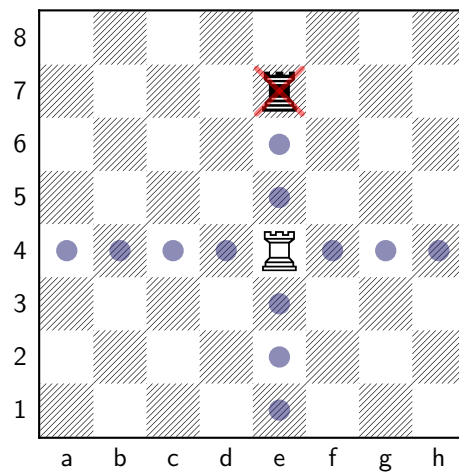


Fonte: Elaborada pelo autor.

Torre

A torre move-se e captura peças apenas nas direções vertical e horizontal, em qualquer sentido, conforme mostra a Figura 3. Se cumpridos os requisitos, a torre pode ser movimentada em um lance especial em conjunto com o rei, chamado roque.

Figura 3 – Movimentação da torre

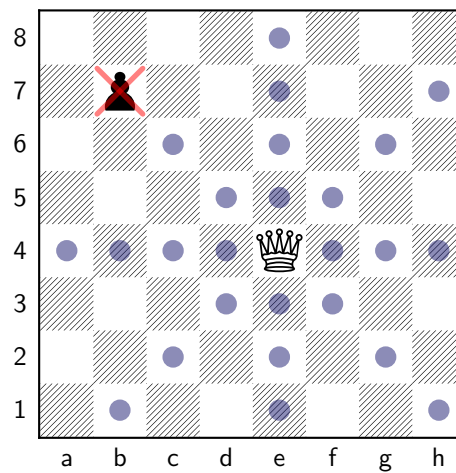


Fonte: Elaborada pelo autor.

Dama

A dama, sendo uma das peças mais poderosas do xadrez, pode ser movida e capturar peças nas direções vertical, horizontal e diagonal, em qualquer sentido, conforme Figura 4.

Figura 4 – Movimentação da dama

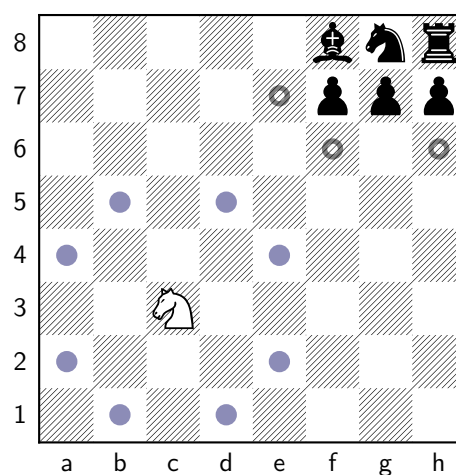


Fonte: Elaborada pelo autor.

Cavalo

O cavalo se move em um padrão de 'L', ou seja, pode mover-se duas casas em uma direção (horizontal ou vertical) e mais uma casa na direção perpendicular, conforme demonstrado intuitivamente na Figura 5. Além disso, ele é a única peça que pode pular sobre outras peças no tabuleiro, o que significa que ele pode se mover para sua casa de destino mesmo que existam outras peças no caminho.

Figura 5 – Movimentação do cavalo



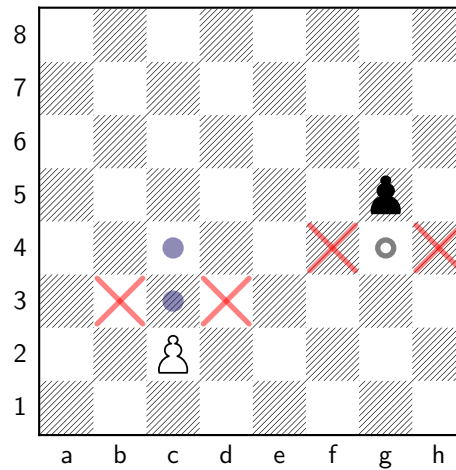
Fonte: Elaborada pelo autor.

Peão

Os peões das brancas movem-se da fileira 2 em direção a fileira 8, enquanto os peões das pretas fazem o movimento inverso, da fileira 7 em direção a fileira 1. O peão

pode andar duas casas apenas na primeira vez em que é movido. Após isso pode andar apenas uma casa e pode realizar a captura de uma peça apenas nas casas imediatamente à sua diagonal, no sentido em que se move, conforme exemplificado na [Figura 6](#). Sob circunstâncias específicas, pode ser promovido a outra peça ou realizar uma captura *en passant*.

Figura 6 – Movimentação do peão

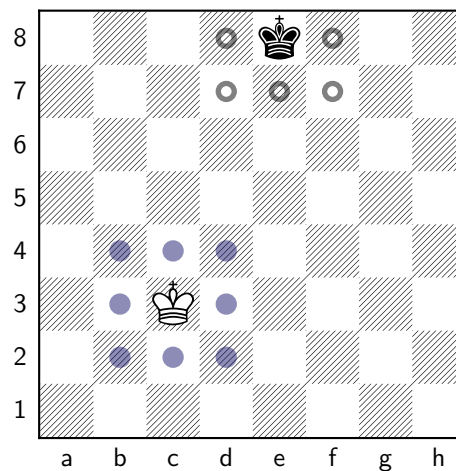


Fonte: Elaborada pelo autor.

Rei

O rei, assim como a dama, movimenta-se e captura peças oponentes em todas as direções. No entanto, ele pode andar apenas 1 casa, conforme demonstra a [Figura 7](#), e não pode colocar-se em xeque.

Figura 7 – Movimentação do rei



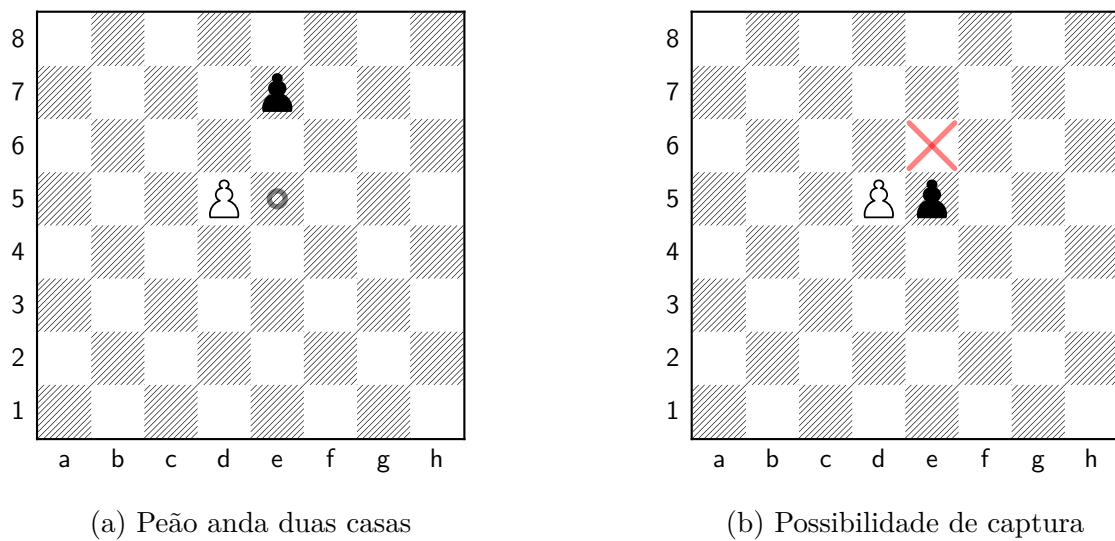
Fonte: Elaborada pelo autor.

2.1.3 Lances especiais

En passant

Um peão que estiver na mesma fileira e em uma coluna adjacente a um peão oponente que acabou de andar 2 casas, pode capturar o peão oponente como se este tivesse andado apenas 1 casa, conforme demonstrado na [Figura 8](#). A captura *en passant* deve ser realizada imediatamente após o peão oponente mover-se duas casas.

Figura 8 – Captura *en passant*



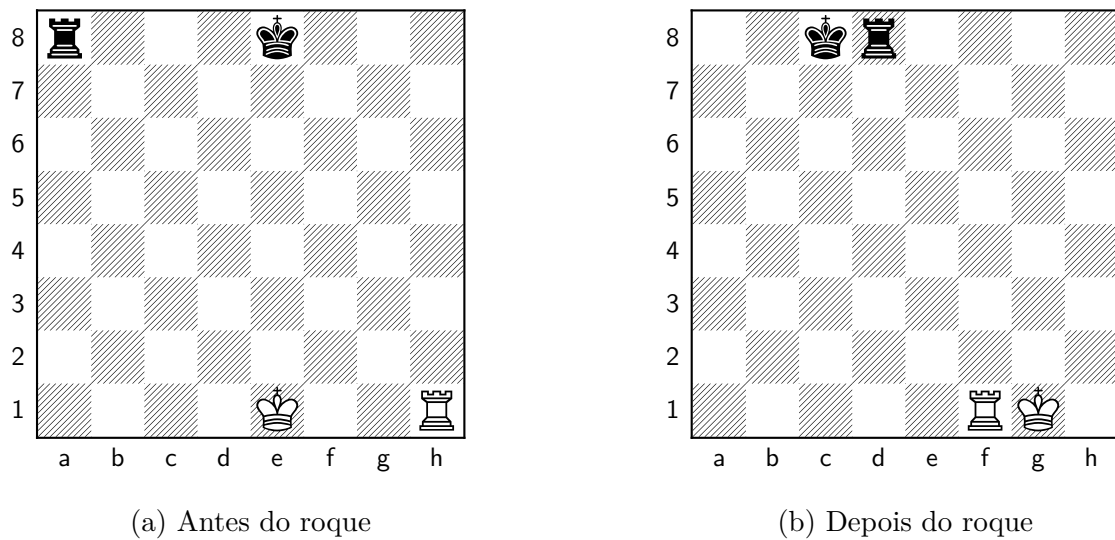
Fonte: Elaborada pelo autor.

Roque

O roque é um lance especial que move duas peças (o rei e uma torre) e conta como um único lance. Como requisitos, ambos o rei e a torre não podem ter sido movidos durante a partida e todas as casas entre as duas peças devem estar desocupadas. Além disso, o rei não pode passar por ou terminar em uma casa sob ataque.

Existem duas variações, o roque longo, realizado com a torre mais distante do rei, e o roque curto, realizado com a torre mais próxima. No roque, o rei anda duas casas em direção à torre, depois a torre move-se para a casa por qual o rei passou. A [Figura 9a](#) mostra uma situação anterior à realização do roque, sendo o roque curto para as peças brancas e o roque longo para as peças pretas, enquanto a [Figura 9b](#) mostra a posição das peças após o roque.

Figura 9 – Roque curto para as brancas e roque longo para as pretas

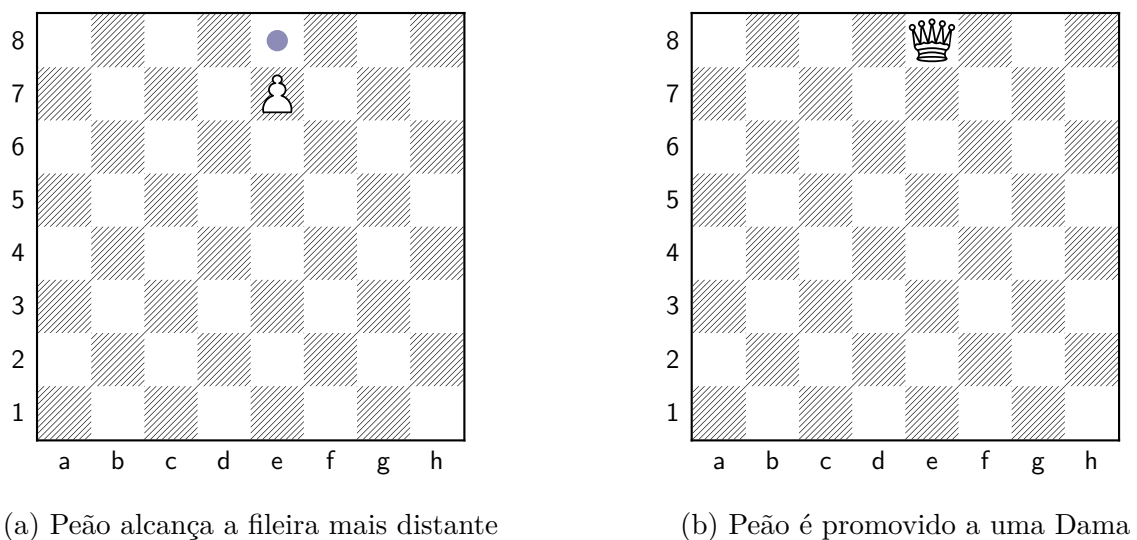


Fonte: Elaborada pelo autor.

Promoção do peão

Quando um peão chega à fileira mais distante de sua casa inicial (fileira 8 para as brancas e 1 para as pretas), ocorre a promoção, em que o peão deve ser substituído, como parte do movimento, por uma nova dama, torre, bispo ou cavalo de mesma cor, conforme demonstrado na [Figura 10](#). A escolha do jogador não é restrita a peças capturadas anteriormente (FIDE, 2022).

Figura 10 – Promoção do peão



Fonte: Elaborada pelo autor.

2.1.4 Notação Algébrica

No xadrez, existem padrões a serem seguidos ao registrar jogadas. A FIDE reconhece em seus torneios e partidas apenas um sistema de notação, chamado Sistema Algébrico (FIDE, 2022).

Neste sistema, cada casa do tabuleiro é mapeada por sua coluna e sua fileira, como demonstrado na Figura 11. A casa ‘a1’, por exemplo, refere-se à casa na primeira fileira da coluna ‘a’, enquanto ‘e7’ é a casa na sétima fileira da coluna ‘e’.

Figura 11 – Notação do tabuleiro

8	a8	b8	c8	d8	e8	f8	g8	h8
7	a7	b7	c7	d7	e7	f7	g7	h7
6	a6	b6	c6	d6	e6	f6	g6	h6
5	a5	b5	c5	d5	e5	f5	g5	h5
4	a4	b4	c4	d4	e4	f4	g4	h4
3	a3	b3	c3	d3	e3	f3	g3	h3
2	a2	b2	c2	d2	e2	f2	g2	h2
1	a1	b1	c1	d1	e1	f1	g1	h1
	a	b	c	d	e	f	g	h

Fonte: (OVERLEAF, 2024)

Conforme o Quadro 1, cada peça do xadrez é representada por uma letra, sempre maiúscula, que é utilizada na notação de lances. O registro do movimento de uma peça é composto pela peça movida, seguida pela casa de destino. Para o peão, não é utilizada a letra representativa. Por exemplo, se a Dama for movida de ‘d1’ para ‘g4’, a notação será ‘Qg4’ (Dama se move para ‘g4’), enquanto um peão movido de ‘e2’ para ‘e4’ será anotado apenas como ‘e4’ (Peão se move para ‘e4’).

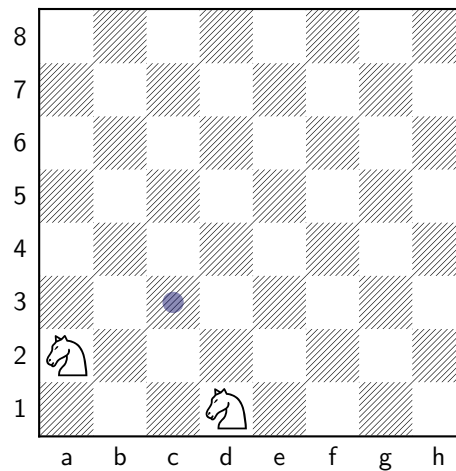
Para jogadas com captura, deve ser adicionada a letra ‘x’ entre a peça movida e a casa de destino. Por exemplo, se uma torre captura uma peça em ‘h5’, a notação será ‘Rxxh5’. Quando um peão faz uma captura, utiliza-se a coluna da qual o peão partiu, seguida pela casa de destino, por exemplo ‘exd5’ (Peão na coluna e captura a peça em ‘d5’). Para capturas *en passant*, utiliza-se a coluna de partida do peão, seguido pelo ‘x’, depois a casa para qual o peão se moveu (não a do peão capturado), seguido pela notação ‘e.p’, porém esta última não é obrigatória.

Em alguns casos, é possível que duas peças iguais e de mesma cor possam mover-se para a mesma casa. Nestes casos, a inicial da peça é seguida pela coluna de partida, se estiverem na mesma fileira, ou pela fileira de partida, se estiverem na mesma coluna.

Quando as duas peças estiverem em fileiras e colunas diferentes, é preferível utilizar a coluna de partida (FIDE, 2022).

Conforme exemplificado na Figura 12, quando dois cavalos podem mover-se para a casa ‘c3’, a notação ‘Nc3’ não é o suficiente para descrever a jogada, pois seria perdida a informação sobre qual cavalo moveu-se para a casa de destino. Nesta situação, deve-se indicar a coluna de partida da peça com ‘Nac3’ ou ‘Ndc3’.

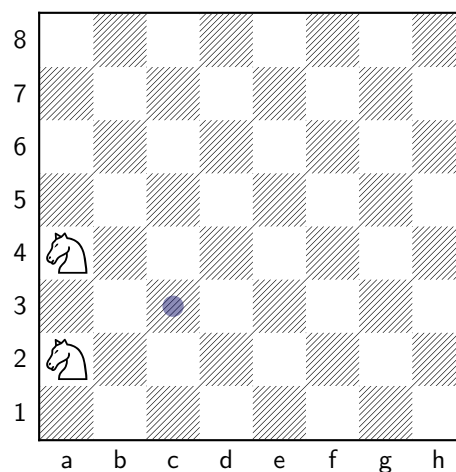
Figura 12 – Desambiguação com cavalo



Fonte: Elaborada pelo autor.

No entanto, existem casos ainda mais críticos para desambiguação, como quando dois cavalos estão na mesma coluna e podem mover-se para a mesma casa, conforme demonstrado na Figura 13. Nestes casos, deve-se utilizar a fileira de partida da peça, resultando na notação ‘N2c3’ ou ‘N4c3’.

Figura 13 – Desambiguação com peças na mesma coluna



Fonte: Elaborada pelo autor.

Quando um peão atinge a última fileira e ocorre a sua promoção, a peça escolhida é indicada após a anotação da jogada, por exemplo ‘d8Q’, quando o peão branco atinge a fileira 8 e é promovido a uma dama, ou ‘exf8N’, quando um peão branco atinge a fileira 8 capturando uma peça e é promovido a um cavalo. Além disso, o roque curto é indicado pela notação especial ‘0-0’, enquanto o roque longo é registrado como ‘0-0-0’ (FIDE, 2022).

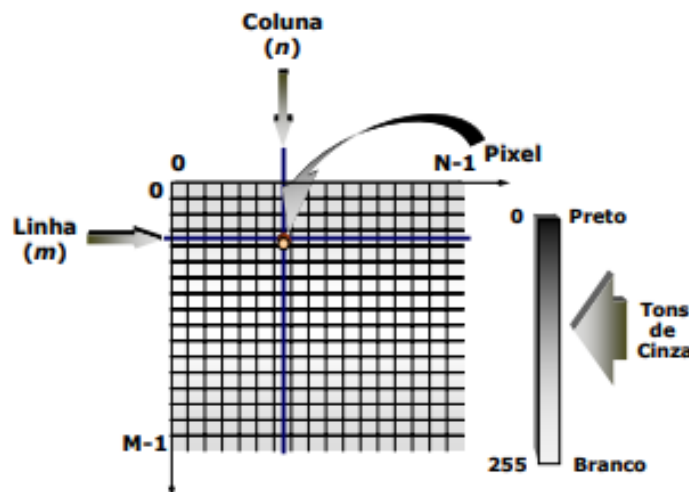
Uma jogada que coloca o rei oponente em xeque deve ter a notação ‘+’ adicionada ao final, enquanto o xeque-mate deve ser indicado como ‘#’ ou ‘++’, embora o primeiro seja mais indicado. A oferta de um empate deve ser anotada como ‘(=)’.

Exemplo do uso da notação algébrica para registrar o xeque-mate do pastor: 1 - e4 e5 2 - Bc4 Nc6 3 - Qh5 Nf6 4 - Qxf7#.

2.2 Técnicas de Processamento de Imagens

Uma imagem é representada digitalmente como uma matriz bidimensional $f(m, n)$ e possui sua origem em $(0, 0)$, no topo à esquerda. Cada elemento desta matriz é chamado *pixel* e, em uma imagem em tons de cinza, pode assumir um valor de 0 a 255, que representa a intensidade luminosa naquele ponto, conforme a Figura 14 (QUEIROZ; GOMES, 2006).

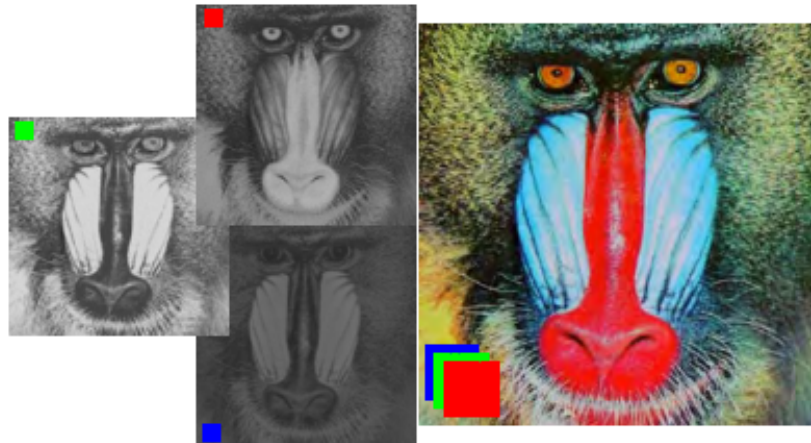
Figura 14 – Representação de uma imagem digital bidimensional



Fonte: (QUEIROZ; GOMES, 2006)

Segundo Gonzalez e Woods (2018), uma imagem digital colorida pode utilizar diferentes modelos de cor, como o RGB (vermelho, verde, azul), CMY (ciano, magenta, amarelo), CMYK (ciano, magenta, amarelo, preto) ou HSI (matiz, saturação, intensidade). Neste trabalho será abordado apenas o modelo aditivo de cores, o RGB.

Figura 15 – Representação de uma imagem digital colorida (modelo RGB)



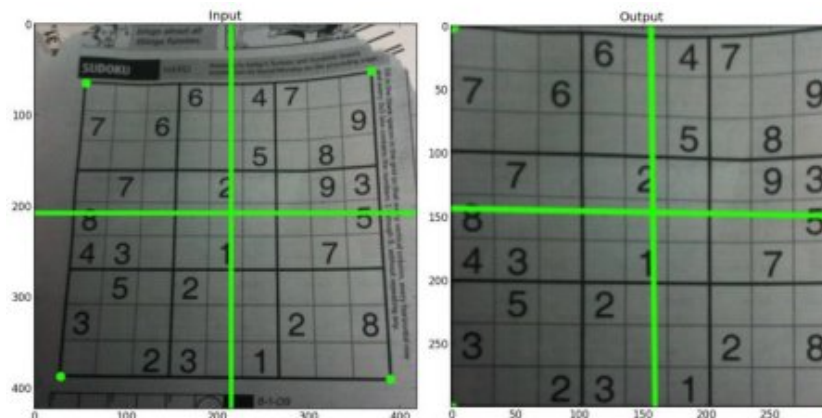
Fonte: (QUEIROZ; GOMES, 2006)

Em um sistema RGB, a imagem colorida é formada pela composição de três imagens monocromáticas em que cada *pixel* representa a intensidade de sua cor. A Figura 15 apresenta os planos monocromáticos de uma imagem e o resultado da composição dos três planos (QUEIROZ; GOMES, 2006).

2.2.1 Transformação de Perspectiva

A transformação de perspectiva é uma técnica usada para corrigir o ângulo de uma imagem, tornando-a similar à visão direta, como se estivesse sendo vista de frente. Essa técnica é utilizada, por exemplo, em aplicativos de escaneamento de documentos, onde as fotos podem ser tiradas em ângulos variados.

Figura 16 – Exemplo de transformação de perspectiva



Fonte: (OPENCV, 2024)

Conforme exemplificado na Figura 16, o processo envolve a definição de quatro pontos na imagem original que correspondem aos cantos da superfície ou objeto a ser ajustado. Esses pontos são então mapeados para uma nova posição em um plano frontal.

A transformação calcula a relação entre esses pontos e ajusta todos os outros *pixels* da imagem de acordo. Essa operação é realizada utilizando uma matriz de transformação que aplica uma série de cálculos para reposicionar cada *pixel* da imagem.

2.2.2 Detecção de Bordas

A detecção de bordas é uma técnica essencial no processamento de imagens, utilizada para identificar os contornos dos objetos presentes na imagem. Segundo Szeliski (2022), bordas são locais onde há uma mudança brusca na intensidade de cor ou luz.

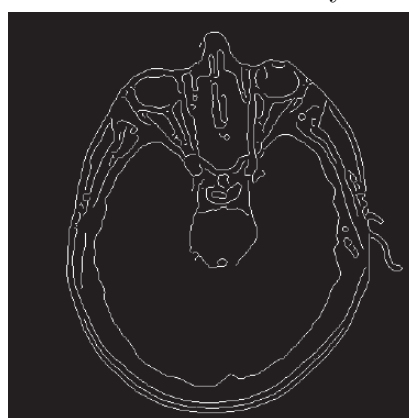
O método de detecção de bordas de Canny (CANNY, 1986) é um algoritmo popular devido a sua alta precisão e baixa taxa de erro, conforme observado na Figura 17. Segundo Gonzalez e Woods (2018), este detector segue uma série de etapas:

- **Redução de ruído:** Aplica um filtro Gaussiano para reduzir ruídos e suavizar a imagem.
- **Cálculo do gradiente:** Encontra a direção e magnitude do gradiente de intensidade da imagem.
- **Supressão de não-máximos:** Procura por valores máximos locais e realiza a supressão de não-máximos para remover *pixels* que não fazem parte das bordas, resultando em linhas mais finas.
- **Histerese com dois limiares:** Utiliza um limiar duplo para determinar se os *pixels* restantes são parte de uma borda.

Figura 17 – Exemplo de aplicação do detector de bordas de Canny



(a) Imagem original



(b) Resultado do detector

Fonte: (GONZALEZ; WOODS, 2018)

2.3 Visão Computacional

A visão computacional é um campo da inteligência artificial cujo objetivo é permitir que máquinas façam decisões úteis sobre objetos físicos e cenas reais com base em imagens capturadas. Segundo [Shapiro e Stockman \(2001\)](#), para tomar essas decisões, é quase sempre necessário construir uma descrição ou modelo dos objetos a partir da imagem, o que muitos especialistas definem como a construção de descrições de cenas a partir de imagens.

Complementando essa definição, [Szeliski \(2022\)](#) destaca que a visão computacional envolve a recuperação da forma tridimensional e a aparência de objetos em imagens, tratando-se de um “problema inverso”. Isso significa que se busca recuperar informações desconhecidas a partir de dados insuficientes, recorrendo a modelos físicos, probabilísticos ou ao aprendizado de máquina a partir de grandes conjuntos de exemplos.

Pode-se entender, então, que a visão computacional é um campo que combina técnicas matemáticas e computacionais para interpretar e entender imagens digitais, reconstruindo propriedades como forma, iluminação e distribuição de cores dos objetos presentes nessas imagens, visando replicar a capacidade humana de percepção visual.

2.3.1 Importância e Aplicações

A visão computacional é essencial em várias áreas, pois permite automatizar e melhorar a análise de imagens e vídeos. Esta tecnologia é utilizada na indústria para a inspeção de máquinas, onde a verificação rápida e precisa de defeitos garante a qualidade e a segurança dos produtos. Em armazéns, a visão computacional automatiza tarefas de embalagem e transporte, aumentando a eficiência e reduzindo custos. Na área médica, é usada para a análise de imagens pré-operatórias e intra-operatórias, além de monitorar a morfologia cerebral dos pacientes ao longo do tempo ([SZELISKI, 2022](#)).

Entre as aplicações mais inovadoras da visão computacional estão os veículos autônomos, que utilizam essa tecnologia para navegar e tomar decisões em tempo real, transformando a forma de deslocamento e aumentando a segurança nas estradas. A modelagem 3D fotogramétrica permite a construção automática de modelos 3D detalhados a partir de fotografias. No entretenimento, a visão computacional é usada em captura de movimento e efeitos visuais, criando experiências cinematográficas imersivas. Além disso, a detecção facial para autenticação visual e a vigilância inteligente são exemplos de como a visão computacional está melhorando o dia a dia, trazendo avanços em segurança e conveniência ([SZELISKI, 2022](#)).

2.3.2 *Machine Learning*

O *Machine Learning* (ML), ou aprendizado de máquina, é uma área da inteligência artificial focada em desenvolver algoritmos capazes de aprender e tomar decisões a partir de dados. Um algoritmo de ML é capaz de melhorar seu desempenho em uma tarefa específica à medida que é exposto a mais dados relacionados a essa tarefa (GOODFELLOW; BENGIO; COURVILLE, 2016). Segundo Szeliski (2022), existem dois tipos principais de aprendizado de máquina:

- **Supervisionado:** O algoritmo é treinado com pares de entradas e saídas corretas. O objetivo é fazer com que o modelo preveja corretamente as saídas para novas entradas. As principais tarefas são classificação (prever categorias) e regressão (prever valores numéricos).
- **Não Supervisionado:** O algoritmo recebe apenas dados de entrada, sem saídas rotuladas. O objetivo é encontrar padrões ou agrupamentos nos dados. Exemplos incluem *clustering* (agrupamento de dados similares) e análise de componentes principais (PCA) para reduzir a dimensionalidade dos dados.

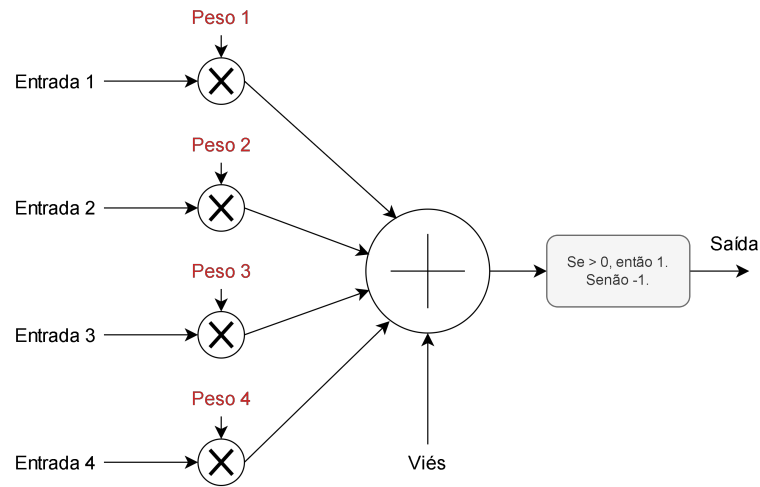
2.3.3 *Deep Learning*

O *Deep Learning* (DL) é um subcampo de *Machine Learning* que utiliza redes neurais profundas, ou *Deep Neural Networks* (DNN), para aprender a partir de dados brutos. Segundo Glassner (2021), os algoritmos de DL, dado treinamento suficiente, são capazes de descobrir automaticamente as regras de tomada de decisão examinando os dados de entrada e extraíndo padrões.

Segundo Glassner (2021), neurônios são a unidade básica de uma rede neural artificial, inspirados nos neurônios reais do sistema nervoso humano, mas de forma muito mais simplificada. Eles recebem dados de entrada, processam essas informações e produzem uma saída.

Em um neurônio moderno, cada entrada pode assumir valores numéricos de ponto flutuante. Esses valores são multiplicados por um peso, que determina a importância de cada entrada. Em seguida, os resultados das multiplicações são somados. Deste valor, é então subtraído o viés, um valor que não vem da saída de outro neurônio. Se o resultado for maior que zero, a saída será 1, senão será -1 (ou 1 e 0), conforme exemplo da Figura 18 (GLASSNER, 2021).

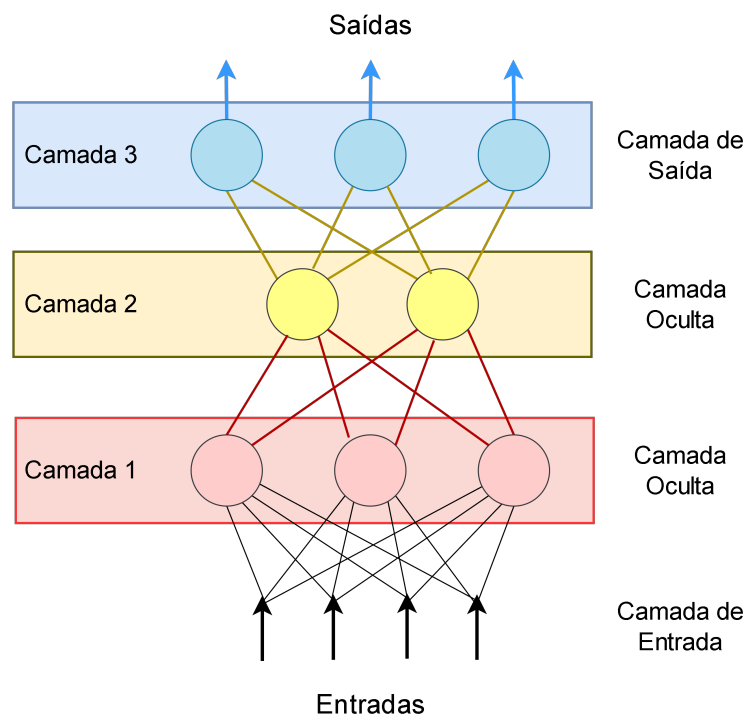
Figura 18 – Funcionamento de um neurônio artificial



Fonte: Adaptado de (GLASSNER, 2021)

Redes neurais artificiais são, como o nome diz, redes formadas pela conexão de neurônios artificiais. Quando estruturadas em camadas, cujos neurônios em uma mesma camada não se comunicam, conforme visualizado na Figura 19, obtém-se uma rede neural profunda, ou DNN. Essas redes neurais profundas são capazes de analisar dados de forma hierárquica, em que cada camada utiliza informações vindas de neurônios da camada anterior para processar maiores conjuntos de dados (GLASSNER, 2021).

Figura 19 – Uma rede neural profunda



Fonte: Adaptado de (GLASSNER, 2021)

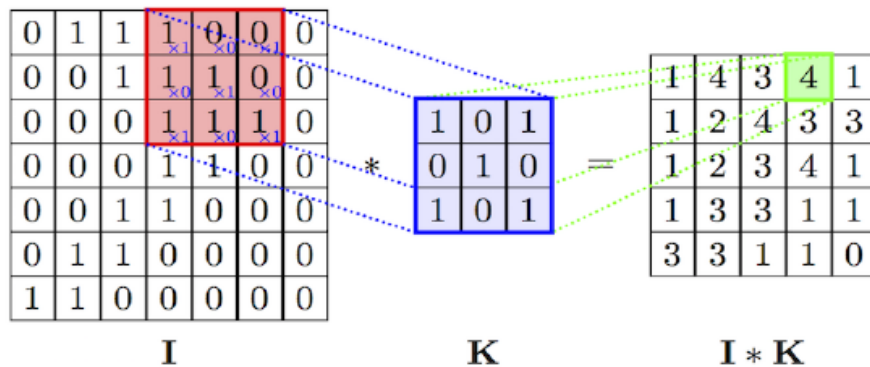
2.3.4 Redes Neurais Convolucionais

Redes neurais convolucionais, ou *Convolutional Neural Networks* (CNN), são um tipo especial de rede neural artificial que utiliza a operação matemática chamada convolução em ao menos uma de suas camadas (GOODFELLOW; BENGIO; COURVILLE, 2016).

2.3.4.1 Convolução

Na visão computacional, a convolução 2D é a mais utilizada. Nessa operação, a matriz de entrada (a imagem) é sobreposta por um filtro chamado *kernel* (uma matriz menor). As duas matrizes são então multiplicadas pixel por pixel, gerando novos valores que são somados para produzir um único valor. O filtro é “deslizado” (*striding*) sobre a imagem, repetindo a operação até cobrir toda a área. Cada valor resultante dessa operação compõe uma nova matriz chamada mapa de atributos (*feature map*), conforme visualizado na Figura 20.

Figura 20 – Exemplo de convolução 2D

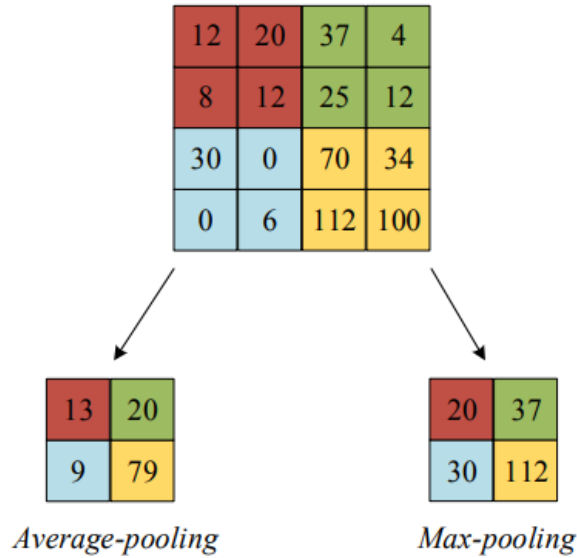


Fonte: (MOHAMED, 2017)

As camadas de convolução possuem como principal função extrair *features*, como bordas, texturas e padrões, mantendo a relação espacial entre pixels e reduzindo o tamanho da entrada de forma controlada, mantendo as informações mais relevantes.

2.3.4.2 Pooling

Outro conceito fundamental das CNNs é o *pooling*, uma forma não linear de subamostragem. Entre as diversas funções de *pooling*, as principais são a *max-pooling* e *average-pooling*. Nesta camada, aplica-se um outro filtro, de forma que não haja sobreposições, conforme Figura 21. Este filtro escolherá o maior valor (*max-pooling*), ou calculará a média dos valores (*average-pooling*), gerando como resultado uma matriz de menor dimensão (MOHAMED, 2017).

Figura 21 – Exemplo de *pooling*

Fonte: (MOHAMED, 2017)

A principal função dessa camada é diminuir a dimensão da entrada, “resumindo” informações das *features* extraídas nas camadas de convolução e, por consequência, reduzir a quantidade de parâmetros e custo computacional da rede neural.

As camadas de convolução e *pooling* podem ser repetidas múltiplas vezes em sequência, a fim de extrair *features* maiores da imagem de entrada, aumentando a robustez da rede.

2.3.4.3 Camada Totalmente Conectada

Após múltiplas camadas de convolução e *pooling*, os *feature maps* são então convertidos em um vetor unidimensional e passados como entrada de uma rede neural, em que cada neurônio está conectado a todos os neurônios da camada anterior. Essa conexão densa forma uma camada totalmente conectada, que permite que a rede combine as *features* extraídas para formar padrões mais complexos (MOHAMED, 2017).

2.3.5 You Only Look Once (YOLO)

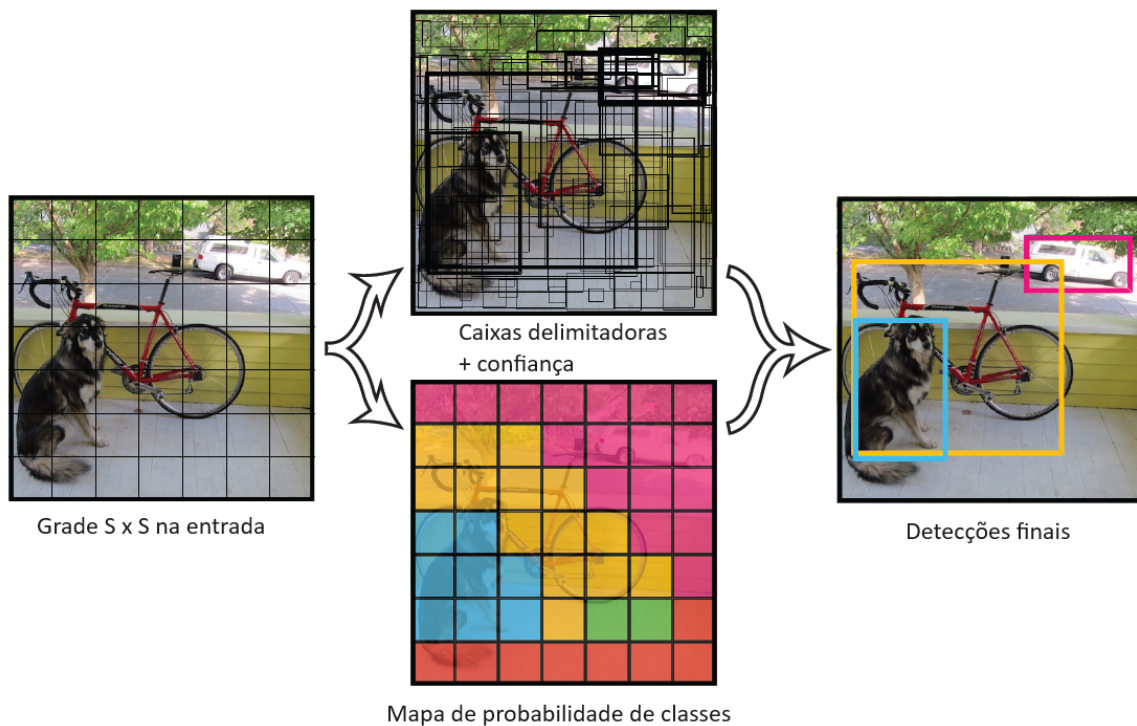
O *You Only Look Once (YOLO)* é uma técnica do estado-da-arte para detecção de objetos voltado a aplicações em tempo real. Ele trata a detecção de objetos como um problema de regressão única e aplica uma única rede neural à imagem. A rede neural divide a imagem em regiões e prevê as coordenadas das caixas delimitadoras e a probabilidade de classes em uma única avaliação para prever quais objetos estão presentes e onde estão (REDMON et al., 2016).

Conforme visualizado na Figura 22, o sistema divide a imagem de entrada em uma grade $S \times S$. Se o centro de um objeto estiver em uma célula da grade, esta célula será a responsável por detectar esse objeto. Cada célula prevê B caixas delimitadoras e gera uma pontuação de confiança, que reflete o quão confiante o modelo está de que aquela caixa contém um objeto.

Cada caixa delimitadora prevê as coordenadas do seu centro em relação aos limites da célula, sua altura e largura em relação à imagem completa e a sua confiança. Cada célula da grade prevê apenas C probabilidades de classe, independente da quantidade de caixas delimitadoras presentes.

A pontuação de confiança e a probabilidade de classe são combinadas para obter a probabilidade de cada caixa incluir um objeto de um tipo específico.

Figura 22 – Funcionamento do YOLO



Fonte: Adaptado de (REDMON et al., 2016)

O YOLO pode ser utilizado como um modelo pré-treinado ou ser treinado do zero. É possível encontrar versões mais atuais do YOLO pré-treinadas com o conjunto de dados COCO¹. Além disso, é possível realizar um ajuste fino do modelo através do treinamento com um conjunto de imagens personalizado.

¹ <https://cocodataset.org>

3 METODOLOGIA

Neste capítulo é apresentada a metodologia a ser utilizada no desenvolvimento do trabalho, incluindo etapas do processo e ferramentas que serão utilizadas.

3.1 Estudos sobre o Xadrez e Visão Computacional

É necessário realizar um estudo sobre o xadrez, a fim de conhecer as regras do jogo e definir o que deverá ser abordado no trabalho, e sobre a visão computacional, reconhecimento de imagem e detecção de objetos, a fim de descobrir possíveis desafios a serem enfrentados na etapa de reconhecimento do tabuleiro e das peças de xadrez presentes no jogo.

3.2 Coleta e Preparação de Imagens

A criação e preparação de um conjunto de imagens é uma etapa essencial quando se trabalha com reconhecimento de imagem utilizando métodos de aprendizado de máquina e visão computacional.

O conjunto de imagens será criado através da coleta de imagens de partidas de xadrez e a acurácia do modelo treinado dependerá diretamente das condições de iluminação do ambiente, além das características do tabuleiro e peças utilizados.

Cada imagem deverá ser preparada manualmente, adicionando marcações que identifiquem o tipo de cada peça, a fim de que o modelo seja treinado para reconhecê-las corretamente.

3.3 Testes do modelo treinado

Após a preparação do conjunto de imagens e treinamento do modelo, deverão ser realizados testes utilizando um subconjunto de imagens, não utilizadas no treinamento, a fim de validar a acurácia do modelo na detecção de cada tipo de peça e sua posição, além dos lances realizados.

3.4 Desenvolvimento e ferramentas

O conjunto de imagens pode ser criado utilizando ferramentas para anotação e desenhar caixas delimitadoras, como o LabelStudio¹, RoboFlow², *Computer Vision Annotation Tool (CVAT)*³ ou outras ferramentas similares. Já o treinamento do modelo pode ser realizado com bibliotecas para a linguagem Python como a Ultralytics⁴ ou plataformas em nuvem como a RoboFlow.

O desenvolvimento computacional do trabalho será realizado com a linguagem de programação Python, com auxílio de bibliotecas e tecnologias para visão computacional e processamento de imagens, como *Open-Source Computer Vision (OpenCV)* e *YOLO*, além do *framework* Kivy para desenvolvimento da interface do aplicativo móvel e a biblioteca *python-chess*⁵ para auxiliar com a validação de lances e outras funções relacionadas ao xadrez.

¹ <https://labelstud.io/>

² <https://roboflow.com/>

³ <https://www.cvat.ai/>

⁴ <https://docs.ultralytics.com/modes/train/>

⁵ <https://python-chess.readthedocs.io/en/latest/>

4 PROPOSTA

Neste capítulo é descrita a proposta do projeto a ser desenvolvido, bem como suas principais etapas e funcionamento, além do cronograma planejado para o desenvolvimento do trabalho.

4.1 Preparação do Conjunto de Dados e Treinamento do Modelo

Inicialmente, serão coletadas imagens de um tabuleiro de xadrez a fim de montar um conjunto de imagens para treinar o modelo de rede neural.

As imagens deverão ser capturadas de diferentes ângulos e sob diferentes condições de iluminação. A escolha do tabuleiro é importante, tendo em vista que alguns tipos de tabuleiro podem ter pouco contraste com as peças e sofrer mais com a reflexão da luz. Além disso, deve-se observar a centralização das peças nas casas do tabuleiro. O conjunto deverá conter diferentes cenários possíveis durante uma partida de xadrez, tanto com peças isoladas quanto com várias peças próximas umas às outras.

O conjunto de imagens construído deverá ser separado em um subconjunto para treinamento, um para validação e outro para testes. Para todo o conjunto, será aplicada a transformação de perspectiva e deverá ser feita a classificação manual dos objetos contidos em cada imagem. A qualidade do conjunto de dados e a correta classificação das imagens possuem grande importância para a precisão do modelo treinado.

4.2 Descrição do Funcionamento da Aplicação

A aplicação deverá receber imagens de uma partida de xadrez como entrada, realizar seu processamento e, como saída, deverá retornar ao usuário o registro completo da partida de acordo com o padrão de anotação da [FIDE](#). O seu funcionamento pode ser separado em diferentes etapas, descritas a seguir, cada uma com sua relevância para a obtenção do resultado esperado.

4.2.1 Detecção e Mapeamento do Tabuleiro

Utilizando-se de técnicas de processamento de imagem, o programa deverá isolar o tabuleiro do restante do cenário na imagem e realizar a transformação de perspectiva, a fim de obter um melhor ângulo e deverá, através da detecção de bordas, encontrar as coordenadas das casas e mapear o tabuleiro de acordo com a notação algébrica de xadrez.

4.2.2 Reconhecimento das Peças Presentes

Em seguida, através da detecção de objetos com **YOLO**, será feito o reconhecimento das peças presentes no tabuleiro, mapeando suas posições e armazenando-as em memória para uso posterior. Ainda, poderá ser utilizada uma biblioteca que permita prever todos os movimentos possíveis para a próxima jogada e armazenar as possibilidades em memória.

4.2.3 Reconhecimento do Lance Realizado

Após receber uma nova imagem, a aplicação irá comparar a nova posição das peças no tabuleiro com aquelas armazenadas anteriormente em memória e, a partir disto, determinar qual foi o lance realizado pelo jogador. Ainda, poderá ser validado o reconhecimento através da comparação do lance reconhecido pelo modelo com a lista de possíveis lances previstos anteriormente.

4.2.4 Geração da Notação Algébrica de Xadrez

Por fim, o programa irá retornar ao usuário todo o registro da partida de xadrez, com todos os lances dos jogadores, em notação algébrica de xadrez e seguindo as normas da **FIDE**, conforme explicado na [subseção 2.1.4](#).

4.3 Cronograma de Atividades

O [Quadro 2](#) descreve o planejamento das atividades para o desenvolvimento deste trabalho.

Quadro 2 – Cronograma de Atividades

Mês	Atividade
Agosto/2024	- Coleta de imagens
Setembro/2024	- Coleta de imagens - Aplicação de técnicas de processamento de imagens
Outubro/2024	- Classificação das imagens - Treinamento do modelo
Novembro/2024	- Desenvolvimento computacional
Dezembro/2024	- Desenvolvimento computacional - Testes da aplicação
Janeiro/2025	- Análise de resultados - Escrita do relatório
Fevereiro/2025	- Preparação da apresentação - Apresentação final

REFERÊNCIAS

- CANNY, J. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8, p. 679 – 698, 12 1986. 20
- FIDE. *FIDE LAWS OF CHESS*. 2022. Disponível em: <https://handbook.fide.com/chapter/E012023>. Acesso em: 20 maio 2024. 9, 10, 15, 16, 17, 18
- FORBES. *From Healthcare To Space: Top 10 Transformative Computer Vision Trends In 2024*. 2023. Disponível em: <https://www.forbes.com/sites/bernardmarr/2023/09/26/from-healthcare-to-space-top-10-transformative-computer-vision-trends-in-2024/?sh=2138bcfd72c0>. Acesso em: 5 abr 2024. 7
- GLASSNER, A. *Deep Learning: A Visual Approach*. [S.l.]: No Starch Press, 2021. ISBN 9781718500723. 22, 23
- GONZALEZ, R. C.; WOODS, f. e. R. E. *Digital Image Processing*. [S.l.]: Pearson, 2018. 18, 20
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. Disponível em: <http://www.deeplearningbook.org>. 22, 24
- HEATH, D.; ALLUM, D. The historical development of computer chess and its impact on artificial intelligence. *AAAI Technical Report WS-97-04*, p. 63–68, janeiro 1997. Disponível em: <https://cdn.aaai.org/Workshops/1997/WS-97-04/WS97-04-013.pdf>. 7
- MOHAMED, I. S. *Detection and Tracking of Pallets using a Laser Rangefinder and Machine Learning Techniques*. Dissertação (Mestrado), 2017. 24, 25
- MURRAY, H. J. R. *A History of Chess: The Original 1913 Edition*. [S.l.]: Skyhorse, 2015. ISBN 9781632207708. 7
- OPENCV. *Geometric Transformations of Images*. 2024. Disponível em: https://docs.opencv.org/4.x/da/d6e/tutorial_py_geometric_transformations.html. Acesso em: 6 agosto 2024. 19
- OVERLEAF. *Chess Notation*. 2024. Disponível em: https://www.overleaf.com/learn/latex/Chess_notation. Acesso em: 24 julho 2024. 16
- QUEIROZ, J.; GOMES, H. Introdução ao processamento digital de imagens. *RITA*, v. 13, p. 11–42, 01 2006. 18, 19
- REDMON, J. et al. *You Only Look Once: Unified, Real-Time Object Detection*. 2016. Disponível em: <https://arxiv.org/abs/1506.02640>. 25, 26
- SHAPIRO, L.; STOCKMAN, G. *Computer Vision*. [S.l.]: Prentice Hall, 2001. 21
- SZELISKI, R. *Computer Vision: Algorithms and Applications 2nd Edition*. [S.l.]: Springer, 2022. 20, 21, 22