

# Classificação de classes de armas no jogo Elden Ring utilizando redes neurais convolucionais

Vítor Hugo Santos de Camargo

**Resumo**—Este trabalho apresenta como objetivo a categorização de diversas classes de armas presentes no jogo de ação e RPG (Role-Playing Game) Elden Ring, utilizando de Redes Neurais Convolucionais (CNN) com a finalidade de examinar e classificar imagens dessas armas. Para tanto, foi utilizado uma base de dados composta por imagens coletadas da wiki do Elden Ring, que possui várias imagens separadas por classes, sobre as quais foram aplicadas técnicas de data augmentation, a fim de ampliar a variedade de dados de treinamento e aprimorar a capacidade do modelo de se adaptar a novas informações. A arquitetura escolhida para o modelo foi baseada na VGGNet, mais precisamente na VGG16, por causa de sua eficácia comprovada em classificar imagens. Os resultados indicaram que o modelo alcançou uma acurácia global de 85,18%, demonstrando alta precisão na categorização das classes de armas selecionadas para classificação do jogo. A avaliação da matriz de confusão juntamente com o recall e precisão para cada classe exibiu informações extras sobre como o modelo está se saindo, destacando seus pontos positivos e apontando áreas de possíveis melhorias. Concluiu-se que, o uso de CNNs representa uma abordagem viável e eficaz para a classificação de armas em jogos como o Elden Ring, que contam com uma variedade de armas distintas, mesmo dentro da mesma categoria.

**Index Terms**—Elden Ring, Classificação de Imagens, Redes Neurais Convolucionais, Data Augmentation, VGGNet.

## I. INTRODUÇÃO

**E**lden Ring é um jogo de ação e RPG, lançado em 2022. Foi desenvolvido pela empresa de jogos FromSoftware e dirigido por Hidetaka Miyazaki com ajuda do escritor de fantasia George R. R. Martin. O jogo foi publicado em diversas plataformas internacionalmente. No jogo, jogadores controlam um personagem cuja missão é ir em busca do Elden Ring e repará-lo, para então se tornar o novo Elden Lord [1].

Elden Ring é jogado em terceira pessoa onde os jogadores exploram livremente seu mundo aberto cheio de inimigos e desafios. As seis principais áreas são atravessadas usando o corcel do personagem do jogador, Torrent, como o principal meio de viagem. Masmorras lineares e escondidas podem ser exploradas para encontrar itens úteis ao jogador. Os jogadores podem usar várias classes de armas e feitiços mágicos, incluindo engajamento indireto possibilitado por mecânicas de furtividade [1].

O objetivo desse trabalho é avaliar e classificar imagens pertencentes a algumas dentre as várias classes de armas presentes no jogo de forma que ao final dele, dada uma imagem de uma arma específica presente no jogo como entrada, seja possível classificar em qual classe tal arma pertence de maneira precisa.

Para a classificação foi utilizado Redes Neurais Convolucionais (CNN), que são um tipo de rede neural artificial comumente aplicadas em análise de imagens. Também foi

utilizado o auxílio de bibliotecas que facilitam o trabalho de utilização da mesma, como a biblioteca Keras.

Assim, este trabalho está organizado da seguinte maneira: Na Seção II são apresentados alguns trabalhos relacionados da área de classificação de armas utilizando de CNNs. Na Seção III é apresentado as Redes Neurais Convolucionais (CNNs), de forma que forneça uma visão geral sobre estas e sua importância para a classificação de imagens. Também é detalhada a metodologia adotada neste estudo, como a descrição da base de dados utilizada, o processo de data augmentation para enriquecimento do conjunto de dados, e a arquitetura específica da CNN empregada para classificação das classes de armas. Na Seção IV são apresentados os resultados obtidos com a execução do modelo construído, incluindo a análise da acurácia geral do modelo, a discussão sobre a matriz de confusão gerada e a interpretação dos resultados de recall e precisão para cada classe de arma. Por fim, na Seção V, o estudo é concluído resumindo os principais achados e o impacto do uso de CNNs na classificação de armas no Elden Ring.

## II. TRABALHOS RELACIONADOS

Diversos estudos recentes têm focado na detecção e classificação de armas no contexto real, como principal objetivo o foco na segurança pública. Estes trabalhos visam identificar armas reais em diferentes cenários, contrastando com o objetivo do presente estudo que se dedica à análise de armas num universo de fantasia.

Chaudhary et al. [2] implementaram um sistema de detecção de armas baseado em CNN, com o objetivo de identificar armas de fogo em vídeos de câmeras de segurança. Através da aplicação de diferentes arquiteturas de CNN, como ResNet50, Inception-ResNetV2, VGG16 e MobileNetV2, o estudo identificou o MobileNetV2 como o modelo mais adequado para aplicações móveis, devido à sua combinação de eficiência e precisão, embora não discutirem os valores obtidos, os autores concluíram que seu estudo oferece uma abordagem promissora para a melhoria do tempo de resposta e redução de danos potenciais em ambientes monitorados.

Kanehisa e Neto [3] focaram na aplicação do algoritmo YOLO (You Only Look Once) para desenvolver um sistema de detecção de armas de fogo, demonstrando a capacidade deste modelo em detectar armas de forma eficiente e rápida. O estudo construiu um conjunto de dados específico para treinamento e teste, extraído do Internet Movie Firearm Database (IMFDB), ressaltando a importância de respostas rápidas e precisas das autoridades de segurança em incidentes

envolvendo armas de fogo, o que pode ser crucial para minimizar o impacto desses eventos. Nos testes de classificação de imagem, de 669 imagens testadas, 644 foram corretamente classificadas. Foram observados 427 verdadeiros positivos e 217 verdadeiros negativos. Os resultados incorretos incluíram seis falsos positivos e 19 falsos negativos. As métricas de avaliação resultaram em 95,73% de sensibilidade, 97,30% de especificidade e 96,26% de acurácia.

Kaya, Tuncer e Baran [4] propuseram um modelo para a detecção e classificação de sete tipos distintos de armas, utilizando como base a arquitetura VGGNet. Este modelo mostrou-se superior em termos de precisão quando comparado com modelos consagrados como VGG-16, ResNet-50 e ResNet-101. Alcançando uma acurácia de sucesso de 98,40% dentre as 5214 imagens de armas contidas na base de dados. O modelo proposto além de superar os demais em precisão, também apresentou uma menor taxa de perda, indicando uma abordagem altamente eficaz para a classificação de armas em imagens.

### III. MATERIAIS E MÉTODOS

#### A. CNN

A eficácia de sistemas de classificação é altamente influenciada pela seleção adequada de características ou representações de dados. Falhar na extração eficiente de informações que distinguem categorias dentro dos dados pode comprometer significativamente o desempenho alcançável [5].

O processo de aprendizagem de representações de forma automática surge como um facilitador na extração de informações importantes para a formulação de sistemas classificatórios eficientes. Neste contexto, técnicas de aprendizado profundo, especificamente redes neurais convolucionais (CNNs), destacam-se [5]. Estas redes representam uma variação das redes de perceptrons multicamadas, sendo organizadas em múltiplas camadas que desempenham funções específicas e complementares. A etapa inicial envolve a aplicação de dados a camadas convolucionais, onde filtros ajustáveis são aplicados por neurônios a áreas determinadas das imagens, com cada neurônio vinculado a um conjunto específico de pixels da camada anterior, atribuindo pesos variáveis a estas conexões. Esses pesos, que podem ser visualizados como matrizes, agem como filtros no domínio espacial das imagens, sendo ajustados ao longo do treinamento para otimizar a detecção de padrões relevantes em qualquer parte da imagem [5].

As peculiaridades dos filtros nas camadas convolucionais, como tamanho, passo (stride) e número, são definidas apenas pela arquitetura da rede, sem a necessidade de especificar as características exatas a serem procuradas. Esse mecanismo permite que a rede ajuste os pesos de forma autônoma durante o treinamento para encontrar os valores ótimos que melhor classificam o conjunto de dados de entrada [5].

A arquitetura utilizada para este estudo foi a VGGNet, mais especificamente o VGG16. A arquitetura VGGNet é amplamente reconhecida por sua eficácia em tarefas de classificação de imagens. Essa estrutura se destaca pelo uso de filtros convolucionais pequenos de  $3 \times 3$ , com passo de um pixel, e camadas de pooling de  $2 \times 2$  posicionadas após cada

duas ou três camadas convolucionais [6]. Este design ajuda na redução da dimensionalidade e na quantidade de parâmetros dos mapas de características produzidos por cada convolução. Notavelmente, a VGGNet emprega também filtros de  $1 \times 1$  como transformações lineares do input.

Especificamente, o modelo VGG16 é composto por 16 camadas, incluindo 13 camadas convolucionais e 3 camadas totalmente conectadas, totalizando aproximadamente 138 milhões de parâmetros. [6].

#### B. Base de Dados

A base de dados utilizadas pelo estudo foi obtida na wiki sobre o jogo Elden Ring [7]. A base continha imagens padronizadas em formato e tamanho de cada categoria de classes utilizadas no estudo, sendo necessário apenas etiquetar cada uma conforme sua classe especificamente para análises manuais visuais posteriores. A tabela I demonstra a quantidade de imagens encontradas por classe.

| Número | Classe         | Quantidade |
|--------|----------------|------------|
| 1      | Axe            | 13         |
| 2      | Curved Sword   | 15         |
| 3      | Dagger         | 16         |
| 4      | Hammer         | 15         |
| 5      | Katana         | 8          |
| 6      | Spear          | 16         |
| 7      | Straight Sword | 19         |

Tabela I: Imagens por classe da base de dados

#### C. Data Augmentation

O uso de redes neurais requerem uma imensa amostra de treinos para obter uma boa performance [5]. Porém, como demonstrado na tabela I, o número de imagens por classe encontradas era bem pequeno, portanto, fez-se necessário a utilização de técnicas como data augmentation para ampliar a quantidade de imagens disponíveis para treinamento e melhorar o desempenho do modelo. A data augmentation é uma técnica amplamente utilizada em aprendizado de máquina para aumentar o tamanho e a diversidade de datasets através da geração de novas imagens a partir das existentes. Isso é feito aplicando uma série de transformações, como rotação, redimensionamento, corte, alteração de cor e perspectiva, entre outras [8]. Exemplos de transformações aplicadas a uma das imagens das classes pode ser visto na figura 1.

A biblioteca utilizada para realizar a data augmentation foi a Keras, mais especificamente a classe ImageDataGenerator, que oferece métodos para facilitar a geração de novas imagens a partir de um conjunto inicial dado de valores de entrada. Essas transformações ajudam a criar um modelo mais robusto e capaz de generalizar melhor para imagens não vistas, compensando assim a limitação inicial de imagens por classe. O método não só introduz um aumento no número de imagens para treinamento do modelo, como também enriquece a base de dados com variações que simulam outros cenários.

#### D. Desenvolvimento do Modelo

O desenvolvimento do modelo de treinamento foi integralmente realizado na linguagem de programação Python,

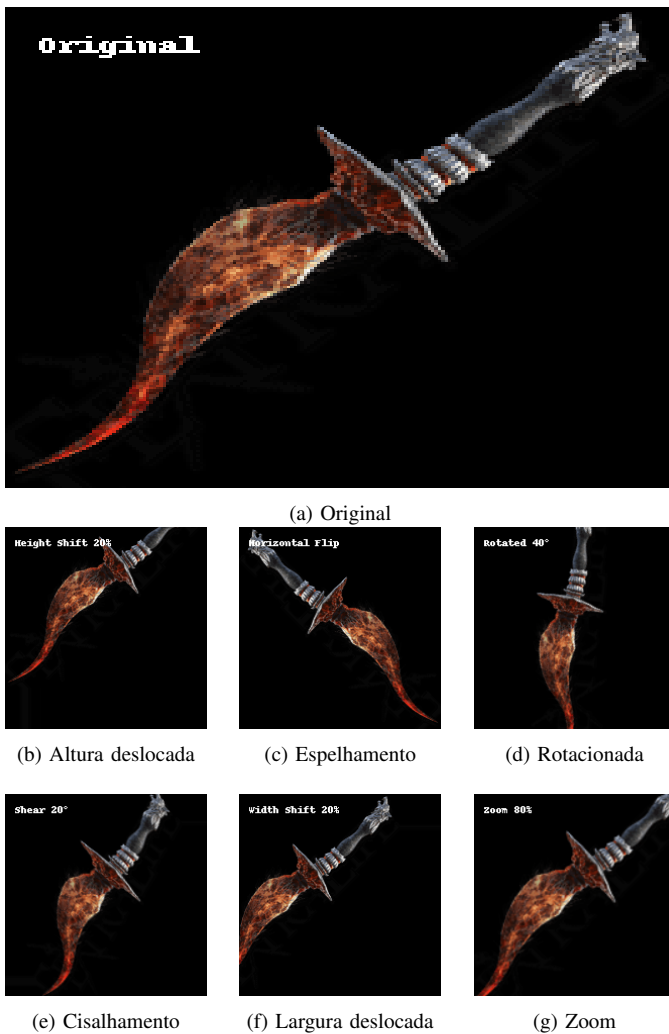


Figura 1: Exemplos de transformações de imagem.

aproveitando-se das capacidades avançadas proporcionadas por bibliotecas do campo de Redes Neurais Convolucionais (CNN). Entre elas, a principal utilizada foi a biblioteca Keras, que opera em conjunto com o TensorFlow, desempenhando um papel tanto no pré-processamento e na augmentação das imagens quanto na implementação do modelo pré-treinado VGG16.

Para a avaliação e validação do modelo, foi utilizado à biblioteca scikit-learn, que possui uma ampla gama de ferramentas estatísticas e de machine learning. Especificamente, as métricas da matriz de confusão e o cálculo do recall e precisão foram empregadas para obter uma compreensão com mais detalhes sobre o desempenho do modelo. Essas métricas são fundamentais para identificar áreas de sucesso e aspectos que necessitam de melhoria.

#### E. Treinamento do Modelo

Inicialmente, as imagens foram divididas em dois conjuntos: o de treinamento e o de teste, onde o conjunto de treinamento possuía em torno de 70% do conjunto total de imagens, enquanto o conjunto de teste os 30% restantes. Este arranjo

é essencial para ensinar o modelo com um conjunto e avaliá-lo com outro, buscando garantir ao máximo que o modelo possa ser aplicado de maneira eficaz a dados novos e não vistos. Como dito anteriormente, o conjunto de treinamento foi enriquecido com técnicas de data augmentation, que aumentam a diversidade dos dados e reduzem a probabilidade de overfitting, um problema que ocorre quando um modelo de aprendizado de máquina se ajusta demais aos dados de treinamento, a ponto de aprender o ruído ou detalhes específicos do conjunto de treinamento, ao invés de capturar padrões generalizáveis que se aplicam a novos dados. Isso pode resultar em um desempenho ruim quando o modelo é aplicado a dados não vistos anteriormente.

Foi inicialmente incorporada a camada pré-treinada da rede VGG16, que atua como base para a extração de características essenciais das imagens. Esta escolha deve-se à capacidade da VGG16 de captar padrões complexos e detalhes visuais em diversas escalas. Todas as imagens precisaram ser padronizadas para o tamanho requerido pela arquitetura para as dimensões  $224 \times 224 \times 3$ . Os pesos inicializados do modelo foram pesos pré-treinados da própria ImageNet, que contém mais de 14 milhões de imagens categorizadas em mais de 20 mil classes diferentes [9]. Portanto, não foi necessário treinar todo o modelo do zero, utilizando assim a técnica de transfer learning, que envolve reutilizar um modelo pré-treinado como ponto de partida para o treinamento de outro modelo. A utilização dessa técnica também ajuda a superar limitações impostas pela base de dados pequena [6].

Após a integração da VGG16, uma série de camadas foram adicionadas para moldar a rede às necessidades específicas da tarefa de classificação em questão. Inicialmente, uma camada de GlobalAveragePooling2D foi utilizada. Essa escolha se deve à capacidade do Global Average Pooling de reduzir as dimensões espaciais da saída da VGG16 para um vetor de características, minimizando o risco de overfitting ao reduzir o número total de parâmetros no modelo [10].

Foi adicionado uma camada densa com 1024 neurônios, empregando a função de ativação ReLU (Rectified Linear Unit). A camada final é uma camada densa com 7 neurônios, correspondendo ao número de classes no conjunto de dados de classificação de espadas. Esta camada utiliza a função de ativação Softmax, que é a mais recomendada para tarefas de classificação multiclasse. A função Softmax transforma os logits - as saídas lineares da última camada densa - em probabilidades, atribuindo a cada classe um valor entre 0 e 1, com a soma de todas as probabilidades resultando em 1. Isso facilita a interpretação das saídas do modelo como confianças associadas a cada classe [11].

Após testes preliminares, o número de épocas foi definido como 25. Uma época corresponde a uma passagem completa pelos dados de treinamento, e o número adequado de épocas é fundamental para evitar tanto o underfitting quanto o overfitting do modelo. A determinação do número ideal de épocas envolve experimentação e observação da performance do modelo em um conjunto de validação ao longo do tempo, não existe um número mágico que funcione para todos modelos.

Fez-se necessário também a utilização de dropout em 0.3, pois também em testes preliminares foi identificado overfitting

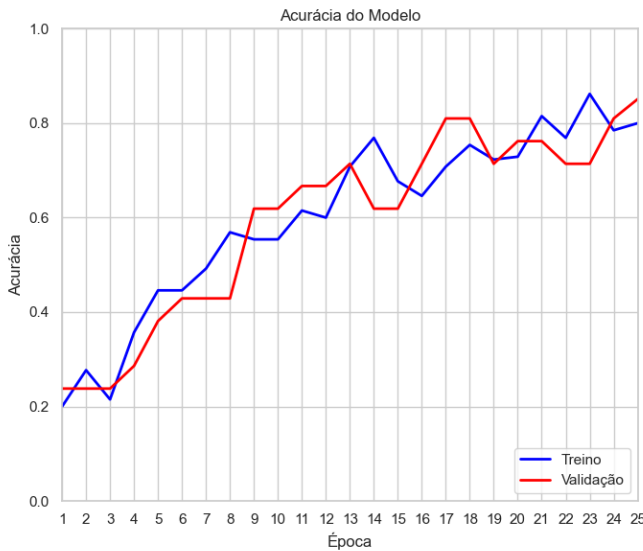


Figura 2: Acurácia do modelo

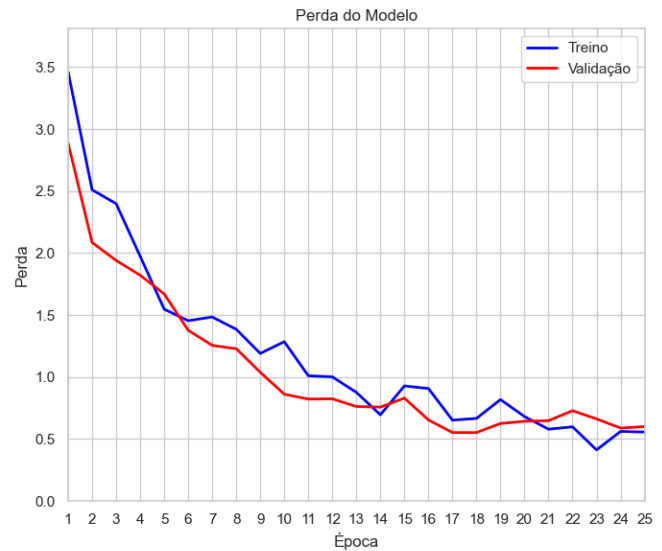


Figura 3: Perda do modelo

no treinamento, onde o modelo aprendia padrões específicos dos dados de treinamento muito bem, a ponto de prejudicar sua capacidade de generalizar para os novos dados.

Para a otimização dos pesos da rede neural, foi empregado o uso do otimizador Adam (The Adaptive Moment Estimation), conhecido por sua eficiência computacional e adequação a dados e parâmetros de grande dimensão. O Adam ajusta adaptativamente a taxa de aprendizagem para diferentes parâmetros, facilitando o ajuste fino do modelo [12]. A taxa de aprendizagem foi configurada para 0.0001, um valor escolhido buscando equilibrar a velocidade do treinamento com a precisão, buscando evitar oscilações ou convergência para mínimos locais.

#### IV. RESULTADOS

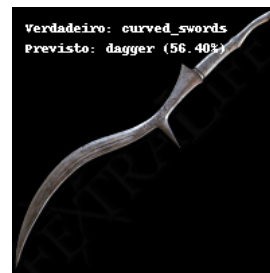
Após o treinamento do modelo, foi possível obter algumas medidas que possibilitam verificar a validação do mesmo, a primeira medida é a acurácia, que é o número total de predições corretas dividido pelo total de predições efetuadas, resultando em 85,18% de acertos, indicando que o modelo possui uma alta taxa de acertos, mesmo em classificações de imagens que possuem similaridade com outras classes, na maioria dos casos.

As figuras 2 e 3 demonstram a acurácia e a perda (loss) do modelo ao longo das épocas. Nota-se que nos dois gráficos os valores de treino e de validação andaram lado a lado, o que sugere que o modelo está conseguindo aprender e generalizar bem e que há pouco ou nenhum overfitting, o que seria indicado pela acurácia de validação acabar diminuindo em alguma etapa em relação ao treino ou, a perda de validação subindo em algum ponto sem acompanhamento da perda de treino.

Outro método de validação utilizado foi a matriz de confusão, demonstrada na figura 8. Percebe-se que de modo geral, o modelo treinado obteve sucesso em classificar as imagens de teste conforme suas respectivas classes. Especificamente

as classes Straight Sword, Spear, Hammer e Axe obtiveram resultados perfeitos, porém, as classes Curved Swords, Dagger e Katana obtiveram alguns resultados errôneos, onde o modelo treinado classificou uma imagem pertencente a classe Curved Swords como Dagger, uma imagem pertencente a classe Dagger como Straight Sword, uma imagem pertencente a classe Dagger como Curved Sword e, uma imagem pertencente a classe Katana como Curved Sword.

Analisando manualmente de maneira visual as 4 imagens classificadas de forma errada pelo modelo, é possível notar que, das 4, 3 imagens são bem similares as imagens da classe na qual foram classificadas erroneamente. As figuras 4, 5, 6 demonstram as comparações entre a classe classificada de forma errada e um exemplo da imagem original da classe. É possível entender as decisões tomadas pelo modelo treinado quanto às decisões devido a similaridade das classes. Já a figura 7 obteve uma diferença discrepante visualmente, o que indica que o modelo ainda pode ser melhorado.



(a) Curved Sword classificada como dagger



(b) Exemplo de uma dagger

Figura 4: Comparação visual entre a imagem classificada como Curved Sword e um exemplo original da classe Dagger

Auxiliando a matriz de confusão apresentada, outra medida de avaliação tomada foi o cálculo de recall e precision apresentados na tabela II. Os resultados indicam que o modelo é capaz de classificar várias classes de espadas com alta precisão



(a) Dagger classificada como Straight Sword



(b) Exemplo de uma Straight Sword

Figura 5: Comparação visual entre a imagem classificada como Straight Sword e um exemplo original da classe Straight Sword



(a) Katana classificada como Curved Sword



(b) Exemplo de uma Curved Sword

Figura 6: Comparação visual entre a imagem classificada como Curved Sword e um exemplo original da classe Curved Sword

e recall. No entanto os valores ainda indicam possíveis melhorias, especialmente na diferenciação de imagens de classes parecidas, como é o caso das figuras 4, 5, 6 e 7 apresentadas anteriormente.

| Classe         | Recall | Precision |
|----------------|--------|-----------|
| Axe            | 100%   | 100%      |
| Curved Sword   | 75%    | 66,7%     |
| Dagger         | 75%    | 75%       |
| Hammer         | 100%   | 100%      |
| Katana         | 66,7%  | 100%      |
| Spear          | 100%   | 100%      |
| Straight Sword | 100%   | 83%       |

Tabela II: Recall e precision de cada classe



(a) Dagger classificada como Curved Sword



(b) Exemplo de uma Curved Sword

Figura 7: Comparação visual entre a imagem classificada como Curved Sword e um exemplo original da classe Curved Sword

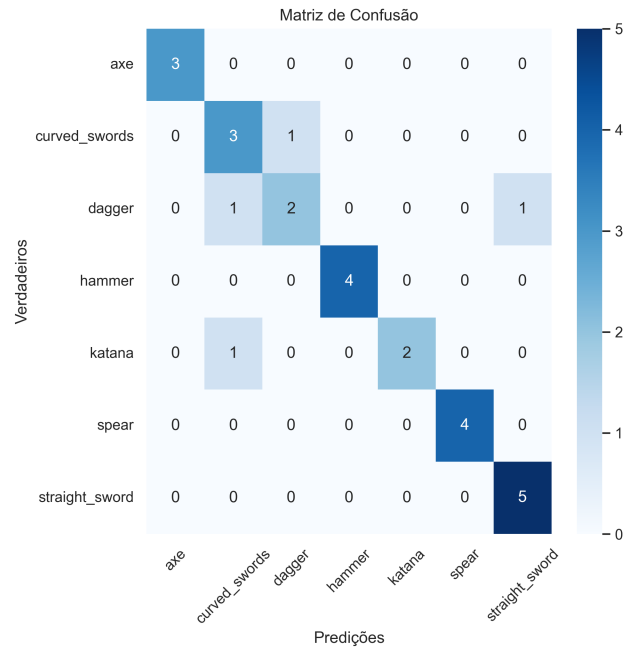


Figura 8: Matriz de confusão do modelo

## V. CONCLUSÃO

Este trabalho demonstrou a eficácia das Redes Neurais Convolucionais (CNN) na tarefa de classificar diferentes classes de armas no jogo Elden Ring. Através da implementação do modelo utilizando a arquitetura VGG16, foi possível alcançar uma acurácia de 85,18%, evidenciando a capacidade da CNN de identificar e diferenciar de maneira eficiente as características visuais das armas pertencentes a diferentes classes. Além disso, a utilização de técnicas de data augmentation provou ser fundamental para enriquecer o conjunto de dados e melhorar o desempenho do modelo em um cenário com uma variedade limitada de imagens para treinamento.

A análise da matriz de confusão e das métricas de recall e precisão ofereceu informações importantes sobre o desempenho específico do modelo em cada classe de armas, identificando áreas em que o modelo demonstrou alta precisão, bem como aquelas que apresentaram espaço para melhorias. As dificuldades encontradas na classificação de algumas classes, particularmente onde houve similaridades visuais marcantes entre classes diferentes, destacam a necessidade de estratégias adicionais de pré-processamento ou de ajustes na arquitetura e valores da rede para aprimorar a distinção entre essas classes, seja pelo número de épocas, taxa de aprendizado ou algum dos outros valores que podem influenciar fortemente os resultados do modelo.

## REFERÊNCIAS

- [1] Wikipedia, "Elden ring." [https://en.wikipedia.org/wiki/Elden\\_Ring](https://en.wikipedia.org/wiki/Elden_Ring). Acessado em: 12 mar. 2024.
- [2] M. A. Chaudhary, S. Sayed, and S. M. Danish, "Weapon detection and classification using convolutional neural network," *Journal of Research in Engineering and Applied Sciences*, vol. 7, no. 3, pp. 382–385, 2022. ISSN (Print): 2456-6411 | ISSN (Online): 2456-6403.

- [3] R. F. d. A. Kanehisa and A. d. A. Neto, "Firearm detection using convolutional neural networks," in *Proceedings of the 11th International Conference on Agents and Artificial Intelligence (ICAART 2019)*, (São Luís, Brazil), pp. 707–714, SCITEPRESS – Science and Technology Publications, Lda, SCITEPRESS, 2019. Federal University of Maranhão (UFMA).
- [4] V. Kaya, S. Tuncer, and A. Baran, "Detection and classification of different weapon types using deep learning," *Applied Sciences*, vol. 11, no. 7535, pp. 1–13, 2021. Accessed: 17 August 2021.
- [5] G. H. Matsushita, A. H. Sugi, Y. M. Costa, A. Gomez-A, C. Da Cunha, and L. S. Oliveira, "Phasic dopamine release identification using convolutional neural network," *Computers in Biology and Medicine*, vol. 114, p. 103466, 2019. Available online 25 September 2019.
- [6] M. H. C. Leme, V. S. Simm, D. R. Tanno, Y. M. G. Costa, and M. A. Domingues, "Stingless bee classification: A new dataset and baseline results," in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, vol. 14469 of *Lecture Notes in Computer Science*, pp. 730–744, Springer, Springer Cham, 2023. CIARP 2023, First Online: 27 November 2023.
- [7] Fextralife, "Elden ring wiki." <https://eldenring.wiki.fextralife.com/Elden+Ring+Wiki>. Acessado em: 15 jan. 2024.
- [8] L. Taylor and G. Nitschke, "Improving deep learning using generic data augmentation," *arXiv preprint arXiv:1708.06020*, vol. v1, pp. 1–14, August 2017. Available online 20 August 2017.
- [9] ImageNet, "Imagenet." <https://www.image-net.org>. Acessado em: 12 mar. 2024.
- [10] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv:1312.4400 [cs.NE]*, vol. arXiv:1312.4400v3, pp. 1–10, 2014. <https://doi.org/10.48550/arXiv.1312.4400>.
- [11] Andrea Perlato, "Cnn and softmax." <https://www.andreaperlato.com/aipost/cnn-and-softmax>. Acessado em: 13 mar. 2024.
- [12] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *3rd International Conference for Learning Representations, San Diego*, 2015. arXiv:1412.6980v9 [cs.LG].