

Full House Peruana

Implementando um sistema de visão computacional para calcular as probabilidades numa partida de Poker

André Luís Araújo de Souza

Instituto de Informática

UFG

Goiânia, Goiás

Hugo Fernandes Silva

Instituto de Informática

UFG

Goiânia, Goiás

Rafael Alves Goiás

Instituto de Informática

UFG

Goiânia, Goiás

Victor Matteus Silva Souza

Instituto de Física

UFG

Goiânia, Goiás

Abstract—Esse artigo apresenta um estudo envolvendo visão computacional na implementação de um sistema cuja finalidade é calcular a probabilidade de vitória de jogadores de Poker em uma partida. O objetivo é fornecer uma ferramenta fácil e eficiente para identificar o possível vencedor em uma mesa de Poker em cada rodada durante todo o jogo através de um programa desenvolvido usando a linguagem *Python* e usando técnicas clássicas da biblioteca *OpenCV* em conjunto de técnicas mais modernas de *deep learning*. O artigo explora o desenvolvimento e a utilização desta ferramenta.

Index Terms—jogo, poker, probabilidade e estatística, visão computacional, redes neurais, *python*.

I. INTRODUÇÃO E REVISÃO BIBLIOGRÁFICA

O *poker* existe desde 1829 e começou em Nova Orleans, com os colonos franceses. O jogo envolvia blefes ou apostas originalmente chamadas de '*Poques*', que eram semelhantes ao *draw poker* de hoje. Em 1837, o *poker* era jogado nos Estados Unidos com 5 cartas por jogador, de um baralho de 20 cartas. Desde então, o jogo se espalhou rapidamente e, pouco tempo depois, o baralho de 52 cartas que conhecemos hoje foi introduzido.

Considerando que o *poker* funciona como um sistema de apostas (sejam elas simuladas ou não), uma partida consiste em rodadas nas quais o jogador tem o objetivo de formar a combinação de cartas mais altas considerando duas as quais iniciam em sua mão, e 5 cartas abertas (compartilhadas) para todos os jogadores na mesa (as quais são viradas a cada rodada). No geral, a cada rodada os jogadores podem escolher entre apostar um valor ou fugir. Com um ou mais jogadores apostando, vence a aposta quem tiver a maior combinação de cartas no fim da última rodada (após todas as cartas compartilhadas terem sido viradas) – como prêmio, o vencedor recebe o total dos valores no centro da mesa (também chamado de pote) depositados pelos outros jogadores.

Apesar do *poker* ser internacionalmente difundido, parte de seus jogadores e muitos dos que querem conhecer (seja assistindo ou jogando) pouco levam em consideração a análise de probabilidade intrínseca à(s) rodada(s), já que ao longo de turnos, sem que haja experiência, inferir quem tem mais probabilidade de vencer uma partida pode ser uma atividade desafiadora sem que haja uma ferramenta auxiliadora. Afinal, o ritmo das rodadas e da troca de turnos pode dificultar que a linha de raciocínio seja mantida ao longo do tempo, sendo

ainda mais complexo caso um telespectador de uma partida ao vivo ou um jogador deseje saber a análise probabilística exata.

Em busca de auxiliar na automatização da análise de probabilidade de vitória e derrota em tempo real durante uma partida de Poker, este trabalho propõe utilizar algoritmos de visão computacional para a detecção de features em cartas de baralho comum, assim como cálculos probabilísticos sobre as features detectadas nas cartas dos jogadores e nas cartas compartilhadas para que seja inferido qual jogador na rodada tem maior chance de vencer a aposta.

Para isso, serviram como base repositórios [1] [2] encontrados no Github os quais utilizam estratégias parecidas para resolver o problema de classificação das cartas, além de outra contribuição em código aberto [3] assim como artigos [4] [5] voltados para o problema de cálculo de probabilidades.

Devido as diferentes modalidades de poker e limite de recursos disponíveis, foram estabelecidas restrições e critérios para os experimentos do projeto (vide III. Metodologia), sendo notável que os quais mais evidenciam natureza austera são: a escolha da modalidade "head2head" (i.e., poker de duas pessoas) e o uso de apenas uma câmera.

Dessa forma, é possível de fato contribuir para o desenvolvimento de uma ferramenta que auxilia tais análises exatas no Poker, análises exatas as quais podem ser desatendidas por boa parte dos jogadores e espectadores caso haja necessidade de priorizar outros raciocínios em meio ao jogo.

II. FUNDAMENTOS TEÓRICOS

A visão computacional é uma área da inteligência artificial que visa capacitar máquinas a entender e interpretar informações visuais, assim como os seres humanos. Esse campo tem experimentado um crescimento significativo nas últimas décadas, devido aos avanços em algoritmos, hardware e técnicas de aprendizado de máquina.

Em um primeiro momento, a etapa inicial consistiu em implementar algoritmos clássicos de visão computacional para detecção e reconhecimento dos números e símbolos presentes nas cartas, sendo assim, foram utilizados algoritmos presentes na biblioteca do *OpenCV* [6] para processamento de imagens, detectando bordas e retirando o fundo, para que então fosse gerado um dataset contendo as imagens de cartas porém com

diferentes planos de fundo para simular distintos casos do jogo.

O objetivo foi extrair as features para que então redes neurais fossem aplicadas (veja a Fig. 1.) –

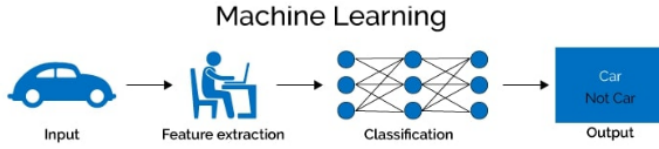


Fig. 1. Fluxograma ilustrativo representativo de como a primeira abordagem funciona. – Machine learning

mas o projeto foi pivotado para a utilização de Deep Learning (veja a Fig. 2.)

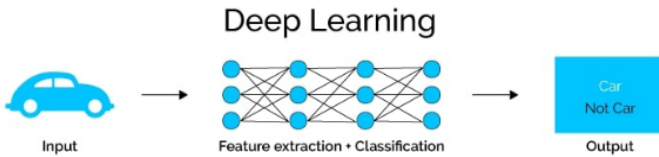


Fig. 2. Fluxograma ilustrativo de como a abordagem para qual o projeto pivotou funciona. – Deep learning

com base na YOLO [7] (You Only Look Once), visto que atualizações da YOLO nos últimos meses trouxeram melhorias no desempenho de modelos de visão em tempo real, e aprimoramento de performance dos modelos (veja a Fig. 3.)

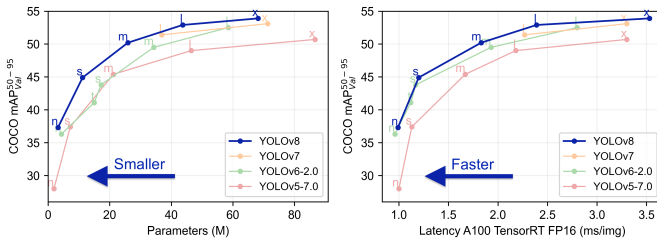


Fig. 3. Comparando diferentes versões da YOLO [10]

Para atender o principal fundamento deste trabalho (vide I. Introdução e Revisão Bibliográfica), com base nas informações obtidas durante a partida, o dataset foi utilizado para treinar uma rede neural YOLO, esta a qual é um dos algoritmos mais populares para detecção de objetos em tempo real, pois consegue alcançar um equilíbrio entre precisão e velocidade.

O processo de treinamento da YOLO envolve a alimentação do modelo com as imagens do dataset, onde cada carta está associada a uma bounding box (caixa delimitadora) e sua respectiva classe (número e símbolo). A rede neural foi ajustada de modo a aprender a localização e classificação das cartas de poker em tempo real, permitindo uma identificação rápida e precisa (levando em conta que a simulação de diferentes cenários para cada carta é um conjunto de dados rotulados o qual melhora assertividade da rede no processo de classificação).

Após segmentação e extração de características (pela YOLO), possibilitando a identificação dos elementos relevantes nas cartas sejam eles naipes, números ou letras (treinamento) foi empregada a detecção em tempo real das cartas na mesa de poker, e portanto, para a identificação das cartas durante o jogo, cada jogador utiliza a câmera fazendo a identificação das cartas da mão, assim como a detecção das cartas na mesa. Com o suporte do software IPWebcam [8] (o qual possui conexão com o OpenCV) e a identificação da YOLO após treinamento, o sistema fornece as informações necessárias para o cálculo da probabilidade de vitória de um jogador em uma partida de poker head2head. Veja a Fig. 4.

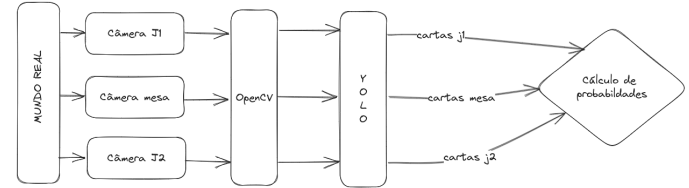


Fig. 4. Fluxograma geral representando o planejamento da solução para calcular probabilidade de vitória numa partida de Poker.

Com essa abordagem, o projeto obtém potencial suficiente para o objetivo, auxiliar jogadores de poker a tomar decisões mais informadas e estratégicas durante suas partidas, assim como espectadores podem considerar análises exatas que, sem esta ferramenta, não são de fácil inferência em meio aos turnos.

III. METODOLOGIA

Para desenvolver o sistema de visão computacional aplicado à identificação de cartas em uma partida de poker, seguiremos as etapas abaixo:

- **Dataset:** Utilizamos um conjunto de dados com imagens de cartas em diferentes posições na mesa e diversas situações de jogo, devidamente rotuladas com informações relevantes (número e símbolo);
- **Implementação de Algoritmos Clássicos:** Usamos o algoritmo de visão computacional OpenCV para conectar com o IP Webcam e captar a imagem para ser testada pelo modelo. —————detecta e reconhecer os números e símbolos nas cartas e através de um pré-processamento, gerar o dataset. lendo o modelo está ativo. clicka na interface.
- **Utilização da Arquitetura YOLO:** Empregamos a arquitetura YOLO Fig. 5) (especificamente o modelo YOLOv8n, o mais "leve" e mais rápido dos cinco modelos disponibilizados pela Ultralytics YOLO (Veja a Fig. 6) para a detecção em tempo real das cartas na mesa. Essa técnica permite uma identificação rápida e precisa.
- **Treinamento da Rede Neural YOLO:** Treinamos o modelo YOLOv8n com o dataset rotulado, ensinando-o a localizar e classificar as cartas em tempo real. Rodamos por 101 épocas utilizando uma GPU NVIDIA A100 pelo Google Colab.

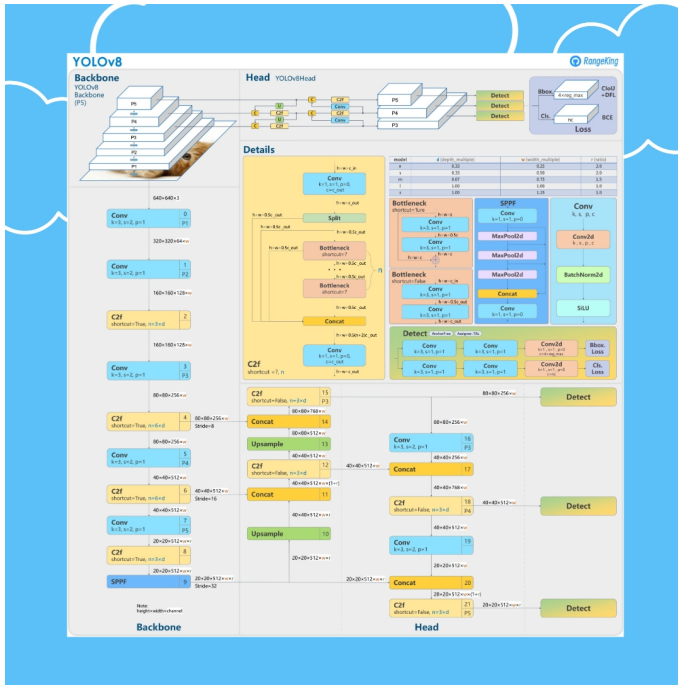


Fig. 5. Arquitetura YOLOv8 [10]

YOLOv8n	YOLOv8s	YOLOv8m	YOLOv8l	YOLOv8x
---------	---------	---------	---------	---------

Fig. 6. Modelos do "mais leve e menos robusto" para "o mais pesado e mais robusto" – última atualização YOLO [10]

- **Cálculo da Probabilidade de Vitória:** Com as cartas identificadas, o sistema calcula a probabilidade de vitória de um jogador utilizando a simulação de Monte Carlo [8] (algoritmo para prever a probabilidade de uma variedade de resultados quando está presente o potencial de variáveis aleatórias) seguindo as regras do Poker sobre hierarquia de combinações (Veja a Fig. 7)
- **Testes e Avaliação:** Realizamos testes em diferentes situações para avaliar a eficiência e precisão do sistema. Alguns exemplos consistem de mãos fracas contra mãos boas, mãos equivalentes e rodadas improváveis, como quando uma mão é fraca no começo e forte no final de uma rodada.

A estratégia aplicada foi com foco em proporcionar um sistema capaz de auxiliar jogadores de poker a tomar decisões mais estratégicas durante suas partidas com base no acompanhamento das combinações possíveis em meio aos turnos.

IV. RESULTADOS E CONCLUSÕES

Depois de desenvolvermos e implementarmos a solução elaborada, testamos algumas vezes para analisar seu funcionamento. Aparentemente o software produzido apresenta um desempenho excelente, para o que se propõe a fazer. Como fora utilizado a Simulação de Monte Carlo para calcular as

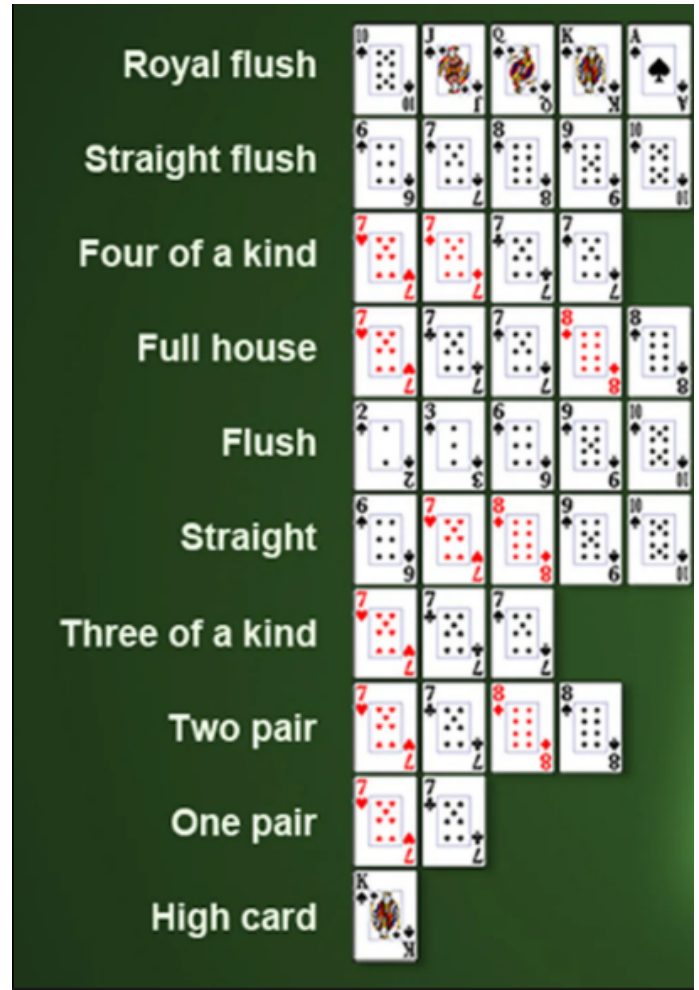


Fig. 7. Regra de hierarquia de combinações de cartas do poker

probabilidades, já não era esperado uma precisão absoluta no resultado das probabilidades. Porém ainda sim os resultados ficaram bem próximos do esperado se fossemos calcular analiticamente tal estatística. Isso nos mostra que, dado a complexidade do problema, conseguimos obter uma solução boa o suficiente para ser utilizada num contexto mais casual. Considerando cenários profissionais, essa solução certamente não conseguira competir com o que é estado da arte hoje em dia, onde os grandes players do mercado usufruem de cartas com sensores e equipamentos milionários para realizar a mesma função que aqui estamos realizando apenas com uma câmera de celular e um computador pessoal.

A. Potenciais aplicações

Após a conclusão do trabalho, se torna relevante a discussão sobre possíveis aplicações dessa ferramenta. Muitos jogadores de Poker iniciantes podem tirar vantagem do sistema que desenvolvemos para conseguir pegar uma intuição de como funciona o jogo mais rapidamente. Além disso, canais de transmissão de partidas oficiais e amadoras podem tirar proveito dessa solução para que o público consiga se divertir mais

enquanto assiste, sem precisar se esforçar mentalmente para entender o que está acontecendo. Ainda no contexto de transmissões de partidas, podemos democratizar esse sistema para qualquer um que almeje transmitir uma partida online, sem ter uma equipe técnica especializada por trás. Por fim, uma outra aplicação bastante pertinente é no campo de pesquisa em Inteligência Artificial, onde essa plataforma poderia ser utilizada para alimentar um modelo de aprendizado por reforço, por exemplo, que aprenderia a jogar poker considerando a possibilidade que ele tem de ganhar a mão, antes de tomar uma decisão.

B. Trabalhos futuros

Diante dos resultados obtidos (os quais podem ser observados no nosso repositório em [10]), trabalhos futuros os quais podem ser realizados envolvem na comparação da performance de modelos da atualização mais recente da YOLO (YOLOv8). Apesar de para este projeto ter sido utilizado o YOLOv8n, há outros modelos que são, por pouco, menos leves e que apresentam bons resultados. Embora iniciamos treinamento com uma YOLOv8s, não houve recursos o suficiente para terminar o treinamento e comparar os resultados dos dois modelos mais leves. Outro trabalho futuro a ser realizado é a implementação de diferentes estratégias probabilísticas para aplicação em tempo real de diferentes modalidades (com enfoque em modalidades que envolvem mais pessoas além das duas em head2head).

REFERENCES

- [1] <https://github.com/geaxgx/playing-card-detection>
- [2] https://github.com/dharm1k987/Card_Recognizer
- [3] <https://github.com/johnsoong216/PokerOddsCalc>
- [4] <https://medium.com/@evantsiklidis/the-essential-math-needed-to-calculate-poker-probabilities-yourself-40ead12079a1>
- [5] <https://blackjackgame.medium.com/texas-holdem-hand-probability-253e0bce7f2e>
- [6] <https://play.google.com/store/apps/details?id=com.pas.webcam&hl=en&pli=1>
- [7] <https://opencv.org/>
- [8] Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
- [9] https://www.wikiwand.com/en/Monte_Carlo_method<https://github.com/ultralytics/ultralytics>
- [10] <https://github.com/cachops/A03TEAM73>