UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Reconhecimento de ações do futebol usando redes neurais convolucionais

João Pedro Fidelis Belluzzo

São Carlos – SP

Reconhecimento de ações do futebol usando redes neurais convolucionais

João Pedro Fidelis Belluzzo

Orientador: **Prof. Dr. Tiago Santana de Nazaré**

Monografia final de conclusão de curso do Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação – ICMC-USP, para obtenção do título de Bacharel em Engenharia de Computação.
Área de Concentração: Ciências de Dados

USP – São Carlos
Outubro de 2022

Belluzzo, João Pedro Fidelis
Reconhecimento de ações do futebol usando redes neurais convolucionais / João Pedro Fidelis Belluzzo. - São Carlos - SP, 2022.
40 p.; 29,7 cm.

Orientador: Tiago Santana de Nazaré.
Monografia (Graduação) - Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos - SP, 2022.

1. redes neurais convolucionais. 2. classificação de eventos. 3. *transfer learning*. 4. esportes. 5. futebol. I. Nazaré, Tiago Santana de. II. Instituto de Ciências Matemáticas e de Computação (ICMC/USP). III. Título.

Dedico este trabalho aos meus pais, meu irmão, meus avôs, avós e à minha namorada.

AGRADECIMENTOS

Agradeço, primeiramente, aos meus pais Ana Silvia e Robson e ao meu irmão João Victor, que sempre me deram apoio e ensinamentos para que eu me tornasse uma pessoa melhor.

Agradeço aos meus avôs e avós - Tilau, Neguita, Antônio e Jacira - por serem verdadeiros exemplos, que sonhavam ver os netos formados e faziam de tudo para ajudar de qualquer forma possível.

Agradeço à minha namorada, Anna Beatriz, por estar ao meu lado em todos os momentos e por me fazer acreditar em mim mesmo.

Agradeço ao meu orientador Tiago por toda atenção e suporte fornecidos durante esse período do projeto, abraçando uma ideia de tema diferente e acreditando em mim desde o início da concepção do trabalho.

Agradeço também à Universidade de São Paulo por toda a estrutura e apoio disponibilizados durante toda a graduação, da qual vou guardar boas memórias.

“Foi o futebol que permitiu uma visão mais positiva e generosa de nós mesmos num plano realmente nacional e popular, como nenhum livro, filme, peça teatral, lei ou religião jamais realizou.” (Roberto da Matta)

RESUMO

BELLUZZO, J. P. F.. **Reconhecimento de ações do futebol usando redes neurais convolucionais**. 2022. 40 f. Monografia (Graduação) – Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos – SP.

A introdução da tecnologia no contexto esportivo se mostra cada vez mais presente, sobretudo no futebol. Entre os tantos pontos de contribuição, entender as ações do esporte de forma automatizada se destaca, haja vista a ascensão da tecnologia VAR (*Video Assistant Referee*). Além das tecnologias já emergentes no esporte, diversos estudos da área abordam estratégias robustas para identificação e classificação dos eventos relacionados ao jogo, como gols, impedimentos, faltas, etc. Em uma perspectiva paralela, é proposta uma abordagem mais simples de ser construída para classificar os momentos desse esporte. A partir de uma base de vídeos de futebol, é utilizada uma rede neural convolucional, a VGG16, para a extração das características dos *frames* que compõem cada evento do jogo. Em seguida, usando uma estratégia de *transfer learning*, é proposta a classificação dos eventos a partir de *ensembles* classificadores: *RandomForest* e *Gradient Boosting* (*LightGBM*). Os resultados, medidos com base na área sob a curva ROC (ROC AUC) e na precisão média (*mAP, mean average precision*), fornecem uma perspectiva sobre o sucesso dessa abordagem adotada, que se mostra diferente daquilo comumente encontrado na literatura da área. É fato que modelos mais robustos apresentam performances de destaque, mas constata-se que a abordagem apresentada neste projeto se mostra um bom ponto de partida para auxiliar na classificação dos eventos do futebol.

Palavras-chave: redes neurais convolucionais, classificação de eventos, *transfer learning*, esportes, futebol.

ABSTRACT

BELLUZZO, J. P. F.. **Reconhecimento de ações do futebol usando redes neurais convolucionais**. 2022. 40 f. Monografia (Graduação) – Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos – SP.

The insertion of technology in sports is increasingly present, especially in soccer context. Among the many contribution points, understanding the actions of the sport in an automated way stands out, as you can see the rise of VAR (Video Assistant Referee) technology. Furthermore, many researches already disscuss robust strategies for recognition and classifications of soccer events, like goals, offsides, fouls, etc. In this work, it's proposed a simplest approach to be built to classify the moments of this sport. From a soccer videos database, it's used a convolutional neural network, VGG16, to extract the features of the frames that form each game event. Then, using a transfer learning strategy, it's proposed the events classification by ensemble methods: RandomForest and Gradient Boosting. The results, measured based on the area under the ROC curve (ROC AUC) and on mean average precision score (mAP), provide a perspective about the success of this approach adopted, which is different from what is commonly found in researches of this area. It is a fact that the more robust models present outstanding performances, but it's noted that the approach introduced in this project can be a good starting point to support soccer events classification.

Key-words: convolutional neural networks, event classification, transfer learning, sports, soccer.

LISTA DE ILUSTRAÇÕES

Figura 1 – Exemplo de convolução entre um tensor $5 \times 5 \times 1$ e um filtro $3 \times 3 \times 1$ . . . 15
Figura 2 – Exemplo de camada densa . . . 17
Figura 3 – Arquitetura simplificada da VGG16. . . . 18
Figura 4 – Classificação por *ensembles*. . . . 18
Figura 5 – Curva ROC em diferentes situações. . . . 19
Figura 6 – Diagrama de blocos genérico para o algoritmo proposto. . . . 21
Figura 7 – Visão geral do algoritmo: (a) extração das características (b) modelo de reconhecimento das ações . . . 22
Figura 8 – Fluxo básico do projeto . . . 25
Figura 9 – Ilustração do algoritmo de geração dos descritores . . . 28
Figura 10 – Exemplo do algoritmo de sintetização das características. . . . 29
Figura 11 – Curvas ROC obtidas para o modelo *LightGBM*. . . . 33

LISTA DE TABELAS

Tabela 1 – Categorias dos eventos da base de dados 27
Tabela 2 – Valores considerados pelo *GridSearchCV* para o *RandomForest*. 30
Tabela 3 – Valores considerados pelo *GridSearchCV* para o *LightGBM*. 30
Tabela 4 – *RandomForest* de melhor performance na base de treinamento. 31
Tabela 5 – *LightGBM* de melhor performance na base de treinamento. 31
Tabela 6 – Precisão por classe e média dos modelos de classificação treinados. 31
Tabela 7 – Precisão média obtida para cada classe em outro projeto da área. 32
Tabela 8 – *Scores* ROC AUC por classe e médio dos modelos de classificação treinados. 33

SUMÁRIO

1 INTRODUÇÃO 12
1.1 Motivação e Contextualização 12
1.2 Objetivos 13
1.3 Organização 13

2 REVISÃO BIBLIOGRÁFICA 14
2.1 Considerações Iniciais 14
2.2 Dados não estruturados 14
2.3 Redes neurais convolucionais 14
2.3.1 Camada convolucional *15*
2.3.2 Camada de pooling *16*
2.3.3 Camada densa ou completamente conectada *16*
2.3.4 VGG16 . . . . . *16*
2.4 Algoritmos de classificação 16
2.4.1 Classificação multi-output *17*
2.4.2 Ensembles *18*
2.4.3 Métricas de performance *19*
2.4.3.1 Área sob a curva ROC (ROC AUC) *19*
2.4.3.2 Precisão média (mAP - mean average precision) *20*
2.5 *Transfer learning* 20
2.6 Trabalhos Relacionados 21
2.6.1 Detecção de eventos do futebol usando aprendizado profundo . . . *21*
2.6.2 Classificação refinada de ações usando redes neurais *22*
2.7 Considerações Finais 22

3 DESENVOLVIMENTO 24
3.1 Considerações Iniciais 24
3.2 Descrição do Projeto 24
3.3 Descrição das Atividades Realizadas 25
3.3.1 Definição da linguagem *25*
3.3.2 Estudo e manipulação das bases de dados *26*
3.3.3 Algoritmo de extração de características (VGG16) *26*
3.3.4 Algoritmo de sintetização das características *28*

3.3.5 *Algoritmo de classificação* *29*
3.4 Resultados Obtidos 30
3.4.1 *Precisão média* *31*
3.4.2 *ROC AUC* *32*
3.5 Dificuldades e Limitações 33
3.6 Considerações Finais 34

4 CONCLUSÃO 35
4.1 Contribuições 35
4.2 Considerações Sobre o Curso de Graduação 35
4.3 Trabalhos Futuros 36

REFERÊNCIAS 37

Glossário 40

Capítulo 1

INTRODUÇÃO

1.1 Motivação e Contextualização

Na história recente, a tecnologia vem se mostrando cada vez mais uma aliada no contexto dos esportes (CAMACHO, 2022). No futebol, especificamente, as evoluções se mostraram ainda mais acentuadas do que nas demais modalidades esportivas. Tais evoluções possuem diferentes públicos-alvo - clubes, jogadores, telespectadores -, mas visam o mesmo objetivo de contribuir com o esporte como um todo.

Para o telespectador, as transmissões de jogos são feitas usando centenas de câmeras, as quais têm se tornado cada vez mais compactas e capturado imagens de maior qualidade. Um exemplo disso é a *SpyCam*, que foi introduzida na Copa do Mundo de 2018 e já se encontra difundida nas transmissões recentes. Tais dispositivos permitem capturar imagens com grande versatilidade de ângulos e de forma instantânea, fator preponderante no esporte (BONFIM, 2018).

Dentro das quatro linhas, diversos são os exemplos. O mais claro deles, sem dúvida, é a utilização do *VAR* (*Video Assistant Referee*), desde 2018 (FIFA, 2022). Porém, os avanços tecnológicos no esporte em si não param aí. Atualmente, os clubes e comissões técnicas investem cada vez mais em análises de desempenho, saúde e performance de seus atletas, fazendo levantamento de diferentes estatísticas que contribuem para as equipes e também para elevar o nível do esporte (ESPORTIVA, 2021).

Nesse contexto, a utilização de diferentes técnicas de análises de dados se faz presente e contribui para coleta e processamento de dados estatísticos. Isso envolve aspectos importantes no gerenciamento de uma equipe, desde o tratamento e prevenção de lesões quanto pelos aspectos do jogo. Análises de grandes quantidades de dados e a determinação de padrões já estão presentes nos estudos táticos das comissões técnicas e também nos núcleos de saúde e performance dos clubes de futebol.

Um aspecto que pode impactar em ambos contextos - dentro e fora de campo - é referente aos momentos do jogo. Identificar ações importantes como faltas, impedimentos, chutes e gols pode ser algo trabalhoso tanto para emissoras quanto para os clubes. Nesse domínio, muitos estudos já utilizam métodos robustos e complexos para identificação e/ou classificação dos eventos do esporte.

Nesse sentido, esse projeto apresenta um estudo sobre o uso de técnicas de ciências de dados para classificar ações em trechos (*highlights*) de um vídeo de um jogo de futebol. Mais especificamente, esse projeto consiste em analisar o uso de redes neurais convolucionais (CNNs – *Convolutional Neural Networks*) como extratores características no contexto de classificação de trechos/lances de jogos de futebol. Trata-se de uma abordagem mais simples de ser construída em relação às demais pesquisas na área, mas com potencial de apresentar uma nova perspectiva sobre o tema.

1.2 Objetivos

Este trabalho tem como objetivo principal estudar o uso de redes neurais convolucionais pré-treinadas para extrair características dos *frames* (imagens) que compõem vídeos de partidas de futebol. Um vez que essas características forem extraídas para cada *frame*, elas serão agregadas de forma a gerar um descritor para o vídeo. Em seguida, modelos serão treinados usando tais características e serão usados para classificar novos lances.

Na literatura atual, muitos autores utilizam técnicas como CNNs 3D (utilizadas em vídeos) (FAKHAR; KANAN; BEHRAD, 2019) e redes neurais recorrentes (SEN *et al.*, 2022) para fazer a classificação dos lances. Tais abordagens têm obtido bons resultados em *datasets* desafiadores, isto é, os modelos de classificação construídos foram capazes de adquirir poder preditivo. No entanto, até onde foi possível verificar, pouco tem sido feito para entender o uso de CNNs 2D (utilizadas em imagens) para realizar tal tarefa. O uso destes modelos pode ser interessante quando se busca, por exemplo, reduzir o custo computacional - energia, memória, processamento, etc. - da classificação de vídeo, dado que elas requerem uma capacidade computacional menor que as CNNs 3D e redes neurais recorrentes.

O presente projeto abordará: a escolha do *dataset* de vídeos de partidas de futebol; a extração das características utilizando uma rede neural convolucional (VGG16); e os modelos de classificação, treinados a partir das *features* extraídas anteriormente. No que diz respeito aos resultados, o objetivo deste projeto é entender se o modelo de classificação criado usando os descritores anteriormente mencionados consegue ter um bom poder preditivo na classificação de lances de futebol.

1.3 Organização

A organização do projeto segue: no Capítulo 2, é dada uma visão sobre como os conteúdos teóricos utilizados são abordados na literatura, além de apresentar brevemente os trabalhos já existentes relacionados ao tema desse projeto; no Capítulo 3, é apresentado detalhadamente todas as atividades do desenvolvimento do projeto, além dos resultados obtidos; no Capítulo 4, é feita a conclusão do projeto, indicando contribuições e possíveis trabalhos futuros.

Capítulo 2

REVISÃO BIBLIOGRÁFICA

2.1 Considerações Iniciais

Nesta capítulo, os conceitos que fundamentam o desenvolvimento do projeto serão apresentados do pontos de vista teórico e bibliográfico. O objetivo é dar uma visão sobre geral sobre tópico estudado e apresentar alguns trabalhos relacionados.

O capítulo trata sobre os seguinte conceitos fundamentas: dados não estruturados; redes neurais convolucionais; a arquitetura da rede neural convolucional VGG16; algoritmos de classificação; curva ROC e além de *transfer learning*. Além disso, serão apresentados dois trabalhos relacionados ao tema do projeto.

2.2 Dados não estruturados

Entende-se por dados não estruturados aqueles que não são arranjados de acordo com algum esquema, modelo ou predeterminação. A exemplo, podemos citar arquivos multimídia, arquivos de texto, *emails*, entre outros. (MONGODB, 2022)

Os dados não estruturados são de extrema importância em aplicações *BigData*. A exemplo disso, segundo MongoDB (2022), estima-se que 90% de todos os dados gerados no mundo de 2016 a 2018 sejam não estruturados.

Esse projeto se relaciona com os dados não estruturados à medida que os objetos de estudo aqui são essencialmente vídeos.

2.3 Redes neurais convolucionais

Uma rede neural convolucional (CNN – *Convolutional Neural Network*) consiste em um algoritmo de aprendizado capaz de receber uma imagem como entrada e gerar alguma previsão sobre a mesma como saída (e.g. previsão de uma classe). Para fazer isso as camadas convolucionais de uma CNN extraem características visuais (i.e. *features*) da imagem. Tais *features* são enviadas para outras camadas (e.g. camadas densas) para serem usadas, por exemplo, para classificar a imagem ou para detectar objetos (IBM, 2020).

Comumente, uma rede neural convolucional é essencialmente composta por arranjos de três camadas:

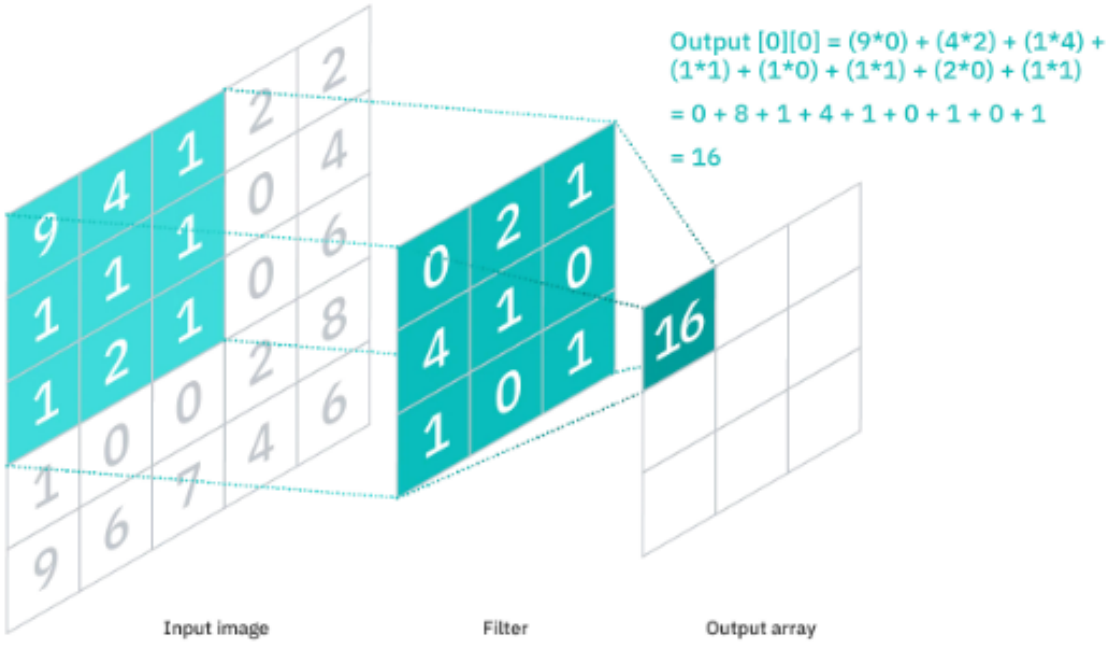
- Camada convolucional
- Camada de *pooling*
- Camada densa ou completamente conectada

2.3.1 Camada convolucional

É a essência de uma rede neural convolucional. A entrada dessa camada é, em muitos casos, um tensor com três dimensões (e.g. uma imagem RGB). A camada possui um filtro, que é iterado sobre os dados de entrada, de forma a gerar uma saída denominada mapa de ativação (IBM, 2020).

Os filtros podem variar de tamanho, mas são tipicamente um tensores $3 \times 3 \times C$, onde C é o tamanho da terceira dimensão do tensor de entrada. O processo de iteração do filtro sobre um tensor segue a operação denominada *convolução*. Nessa operação, é realizado o produto escalar de elementos da mesma posição. A Figura 1 ilustra esse procedimento.

Figura 1 – Exemplo de convolução entre um tensor $5 \times 5 \times 1$ e um filtro $3 \times 3 \times 1$



Fonte: https://www.ibm.com/cloud/learn/convolutional-neural-networks.

2.3.2 Camada de pooling

A camada de *pooling* é responsável por reduzir as dimensões dos dados convoluídos, a partir da aplicação de uma função de agregação (IBM, 2020).

De forma similar à camada convolucional, esta camada também itera sobre as matrizes de entrada, mas dessa vez sem o uso de um filtro. Selecionando um conjunto de dados por iteração, a camada aplica uma função que resume as informações, de forma a diminuir o tamanho em relação à entrada. Essa camada é essencial para reduzir complexidade e aumentar eficiência (IBM, 2020).

Há duas estratégias principais de *pooling*: máximo, em que a saída corresponde ao valor máximo dos dados selecionados; e média (*average*), em que a saída representa a média aritmética dos dados analisados. Segundo Saha (2018), em geral, o *pooling* máximo performa de forma melhor principalmente por descartar ruídos em paralelo com a redução de dimensionalidade e, por isso, foi escolhido para o projeto.

2.3.3 Camada densa ou completamente conectada

Essa camada realiza a classificação das características extraídas a partir das camadas e aplicações de filtros feitas anteriormente.

Utilizando uma função de ativação e um peso, cada elemento dessa camada gera uma saída que representa a entrada do elemento seguinte. Dessa forma, a camada completamente conectada é capaz de classificar uma imagem, conforme as classes de estudo. Um exemplo está mostrado na Figura 2.

2.3.4 VGG16

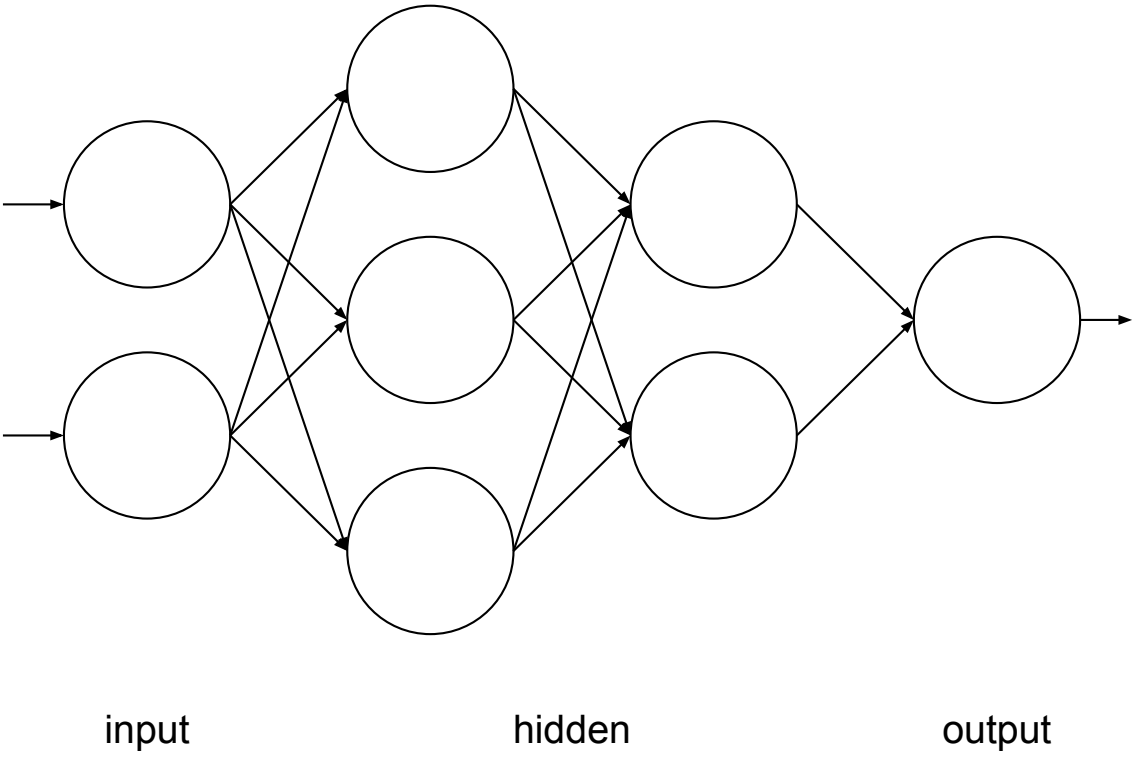
Um exemplo de rede neural convolucional bastante usada em cenários de *transfer-learning* é a VGG16. Utilizando filtros convolucionais de dimensão 3×3, os criadores propuseram uma arquitetura com aproximadamente 138 milhões de parâmetros treináveis. (G, 2021)

A arquitetura padrão dessa rede neural convolucional é composta por 13 camadas convolucionais, 5 camadas de *pooling* e 3 camadas densas, como mostra a Figura 3 e foi a utilizada no projeto.

2.4 Algoritmos de classificação

Os algoritmos de classificação são parte da subárea de aprendizado supervisionado, que consiste em utilizar informações conhecidas para identificar padrões e ser capaz de predizer

Figura 2 – Exemplo de camada densa



Fonte: De autoria própria.

eventos futuros. Nesse caso, o objetivo é adequar em classes os eventos de estudo. (SINGH; THAKUR; SHARMA, 2016)

Experimentos envolvendo esse tipo de modelo normalmente são feitos em duas grandes fases: a primeira consiste no treino do modelo e a seguinte na validação com base em dados de testes usados para medir acurácia e performance.

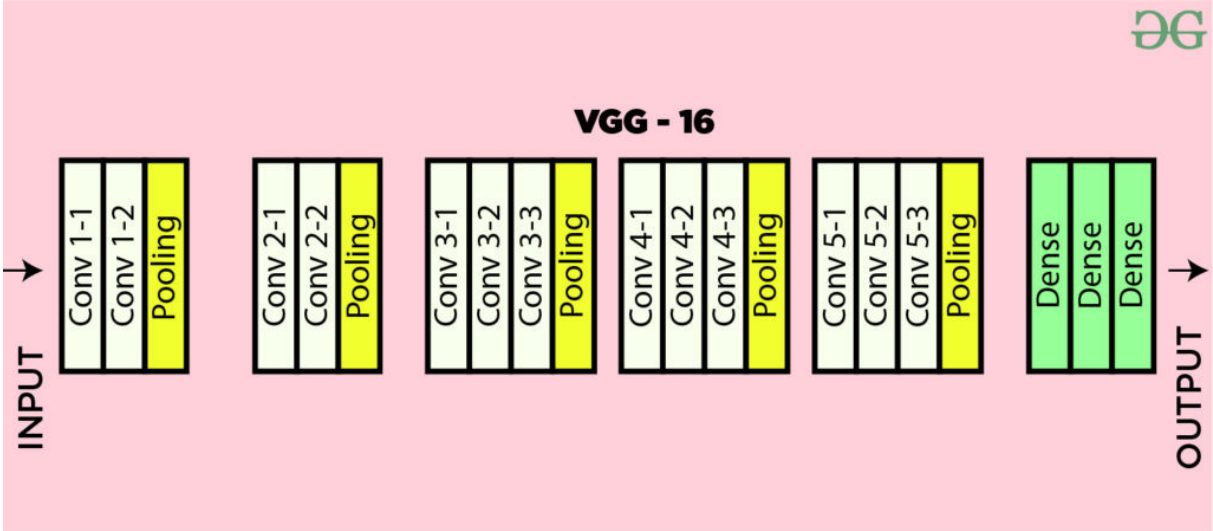
Dentre os exemplos de algoritmos de classificação, podemos citar: as árvores de decisão (*decision trees*), o *Random Forest*, o KNN (*k-nearest neighbors algorithm*) e a regressão logística.

2.4.1 Classificação multi-output

A classificação *multi-output* se trata de uma variação dos problemas de classificação. Nesse tipo de problema, uma instância pode pertencer a uma ou mais classes, ou seja, as classes são não-exclusivas.

Dentre as diferentes técnicas adotados nesses problemas, uma muito utilizada é transformar o problema em uma classificação binária. Isto é, para cada classe é treinado um modelo de classificação binária que indica a existência ou não daquela classe na instância analisada. (NOONEY, 2018)

Figura 3 – Arquitetura simplificada da VGG16.



Fonte: https://www.geeksforgeeks.org/vgg-16-cnn-model/

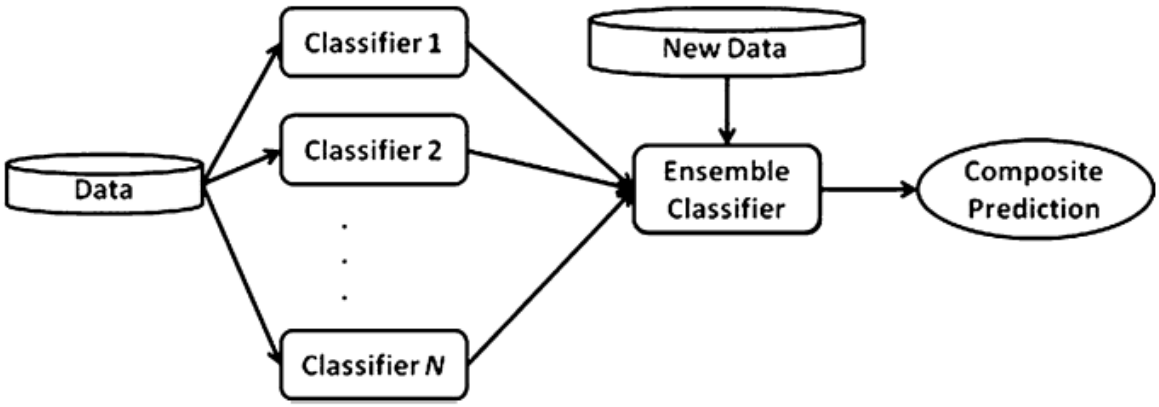
2.4.2 Ensembles

Os algoritmos *ensembles* são capazes de utilizar diferentes modelos simples de aprendizado de forma agregada, com o objetivo de fornecer um resultado final único.

> A ideia da classificação por *ensembles* é aprender não apenas com um modelo de classificação, mas sim com um conjunto deles, e então combiná-los usando alguma forma de votação para classificar uma instância desconhecida. (...) É esperado que os *ensembles* possuam um nível de acurácia preditiva maior que qualquer um modelo classificador individualmente, mas não é garantido. (BRAMER, 2013, tradução nossa)

A Figura 4 ilustra um classificador *ensemble* genérico.

Figura 4 – Classificação por *ensembles*.



Fonte: Bramer (2013).

Como exemplos de *ensembles*, podemos citar as técnicas de classificação *Random Forests* e *Gradient Boosting*.

2.4.3 Métricas de performance

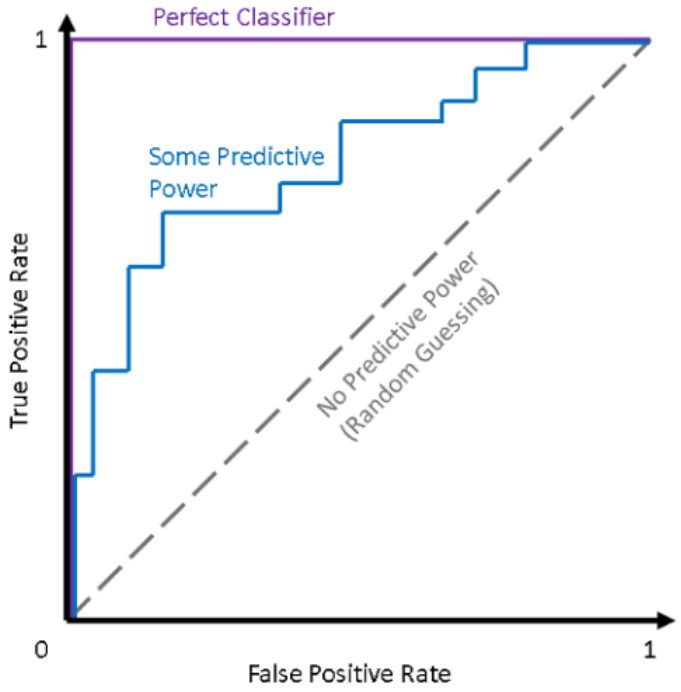
2.4.3.1 Área sob a curva ROC (ROC AUC)

Uma métrica muito utilizada para medir a qualidade das predições de algoritmos de classificação é a área sob a curva ROC (*ROC AUC*). Essa métrica mede a capacidade do modelo em diferenças as classes do problema. A curva ROC em si relaciona a taxa de amostras positivas corretamente detectadas, denominada sensibilidade, ou seja, em que o modelo previu corretamente a existência da classe analisada; e a taxa de amostras falsamente positivas, chamada especifidade, isto é, quando o modelo atribuiu erradamente uma instância negativa à classe positiva.

Ao gerar suas predições, as técnicas de classificação, em geral, são capazes de estimar uma probabilidade para a ocorrência de cada classe. Então, para a construção da curva ROC, são relacionadas a sensibilidade e a especifidade para diferentes limiares de classificação adotados. Com a curva ROC construída, podemos utilizar a área abaixo da mesma como um indicador da performance geral do modelo. A Figura 5 mostra um exemplo de curva ROC.

A área sob a curva ROC é uma métrica que varia de 0 a 1 e indica a capacidade do modelo em identificar a existência ou não de uma classe no problema. Quanto maior, melhor o modelo consegue indicar amostras verdadeiramente positivas. Uma área de 1 (ou 100%) representa um modelo ótimo, enquanto uma área de $0,5$ (ou 50%) indica um modelo aleatório, sem poder preditivo. A Figura 5 indica a curva ROC para diferentes situações.

Figura 5 – Curva ROC em diferentes situações.



Fonte: Steen (2020).

2.4.3.2 Precisão média (mAP - mean average precision)

A métrica de precisão média, ou *mAP* (*mean average precision*), é muito utilizada para detecção de objetos e problemas de classificação. Ela relaciona as medidas denominadas *recall* e precisão.

O *recall* tem o objetivo de indicar o quão bem a classe é identificada pelo modelo, relacionando as amostras corretamente identificadas (sensibilidade) com as amostras não-identificadas, ou seja, que pertenciam a determinada classe e o modelo não conseguiu predizer.

A precisão mede a qualidade das predições, calculando a taxa de amostras que realmente pertencem a uma determinada classe. Isto é, corresponde à divisão das amostras corretamente identificadas pela soma desse mesmo valor com as amostras identificadas como positiva mas de forma errada.

Assim, é possível calcular esses dois parâmetros para diferentes limiares de classificação. A precisão média de cada classe corresponde à soma das multiplicações desses valores em diferentes limiares. Finalmente, o *mAP* refere-se à média das precisões médias de todas as classes.

Essa métrica também varia de 0 a 1. Quanto maior, melhor a capacidade do modelo em identificar corretamente a existência das classes do problema.

2.5 *Transfer learning*

A técnica denominada *transfer learning* consiste em utilizar um modelo pré-treinado para um novo problema e conjunto de dados. Tal técnica é muito utilizada em tarefas de classificação de imagens, principalmente por reduzir a complexidade e facilitar o treinamento dos modelos.

Em suma, podemos praticar o *transfer learning* de três formas, nesse contexto:

- Utilizar o modelo pré-treinado da forma que ele é;
- Utilizar o modelo pré-treinado para inicialização das variáveis do novo modelo;
- Utilizar o modelo pré-treinado para extrair características.

As redes neurais convolucionais, como a VGG16, podem basear seus pesos iniciais no famoso *dataset* chamado *ImageNet*. Trata-se de um grande conjunto de imagens com 1000 classes diferentes. (DENG *et al.*, 2009)

Assim, as redes neurais convolucionais são treinadas em tal base de imagens e seus parâmetros são armazenadas, podendo ser usados em outras tarefas.

Além disso, como apresentado anteriormente, as redes neurais convolucionais possuem uma parte densa, ou completamente conectada, responsável pela classificação da imagem de

entrada. Podemos remover essas camadas da rede neural e utilizar as características extraídas pelas camadas convolucionais e de *pooling*, e então, fazer o *transfer learning* treinando outro algoritmo (e.g. de classificação) para o novo problema.

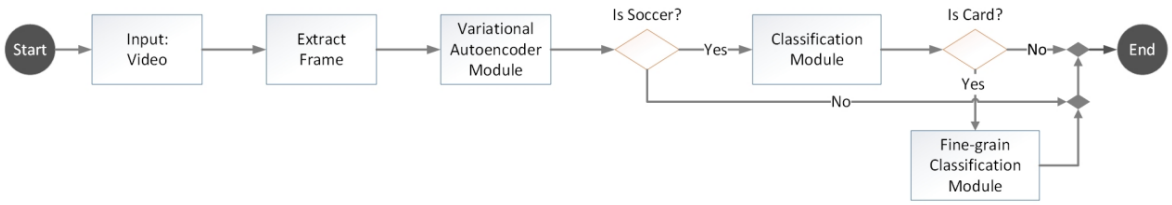
Essas duas estratégias - inicialização dos pesos (variáveis) e a extração de características - contribuem muito para o treinamento dos modelos e são fundamentais para o estudo em questão.

2.6 Trabalhos Relacionados

2.6.1 Detecção de eventos do futebol usando aprendizado profundo

O trabalho Karimi, Toosi e Akhaee (2021) utiliza uma abordagem de aprendizado profundo para detectar e também classificar as imagens extraídas de partidas de futebol. O autor tem o objetivo de, primeiro, detectar momentos relevantes de uma partida e, em seguida, fazer a classificação desses eventos. Uma visão geral do método utilizado está ilustrada na Figura 6.

Figura 6 – Diagrama de blocos genérico para o algoritmo proposto.



Fonte: Karimi, Toosi e Akhaee (2021).

Uma interessante diferença em relação ao escopo abordado nesse artigo reside no fato do autor tratar, inicialmente, de detectar a existência de um evento qualquer, para então classificá-lo. Dessa forma, ele é capaz de configurar um filtro para reduzir os *frames* a apenas aqueles relevantes para o estudo, deixando por exemplo: cartões, pênaltis, chutes, etc.

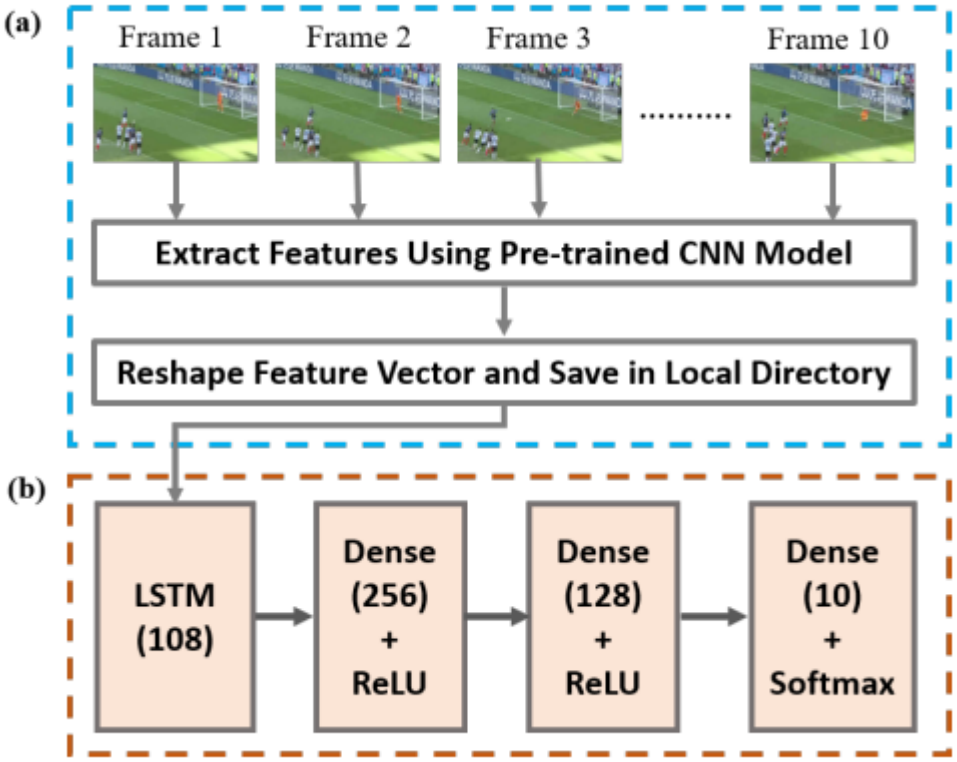
Com os *frames* dos eventos detectados, o autor faz um processo semelhante ao abordado nesse trabalho. Com o apoio de uma rede neural convolucional, ele faz a classificação da imagem conforme as classes definidas previamente. Foi utilizada a *EfficientNetB0* pré-treinada para extração das características, e em seguida, três camadas densas para finalmente determinar a classe do evento.

Finalmente, há um tratamento especial para eventos relacionados a cartões amarelos e vermelhos. Com o objetivo de determinar qual desses eventos está ilustrado especificamente, ainda há um módulo de classificação final entre esses dois eventos, também utilizando redes neurais convolucionais.

2.6.2 Classificação refinada de ações usando redes neurais

A pesquisa Sen *et al.* (2022) utiliza uma abordagem semelhante à desenvolvida nesse projeto para a classificação dos eventos do futebol, conforme mostra a Figura 7. Inicialmente, é realizada a seleção dos *frames* que determinam um evento, seguida da extração das características usando redes neurais convolucionais pré-treinadas a partir da *ImageNet*.

Figura 7 – Visão geral do algoritmo: (a) extração das características (b) modelo de reconhecimento das ações



Fonte: Sen *et al.* (2022).

Em seguida, é feita a classificação do evento. Um ponto interessante nessa parte é que o autor utiliza LSTM (*Long Short Term Memory*), que se trata de uma rede neural que possui memória de curto prazo. Isso é extremamente relevante quando tratamos de vídeos, pois os *frames* subsequentes a serem processados não são independentes uns dos outros. Nesse contexto, carregar informações dos *frames* anteriores pode ser primordial para uma classificação mais precisa dos eventos.

2.7 Considerações Finais

Nesse capítulo, foi dada uma visão geral dos conceitos que fundamentaram o desenvolvimento do projeto. Além disso, pesquisas relacionadas ao mesmo tema também foram brevemente abordadas, dando um panorama de como o assunto já foi tratado em outros trabalhos. A se-

guir, será detalhado o desenvolvimento do projeto, juntamente com os resultados e dificuldades encontradas.

Capítulo 3

DESENVOLVIMENTO

3.1 Considerações Iniciais

Nesse capítulo, será apresentado em detalhes o projeto desenvolvido. A partir de uma visão inicial, será explicada a ideia para o desenvolvimento e apresentado o fluxo básico do projeto. Então, o desenvolvimento de cada etapa será detalhado, com ilustrações e exemplos. Finalmente, serão apresentados os resultados obtidos ao final do desenvolvimento, além das dificuldades e limitações encontradas no processo.

3.2 Descrição do Projeto

O estudo consiste em analisar o uso de CNNs 2D pré-treinadas como descritores de vídeos para aplicações em classificação de lances de jogo de futebol. Para realizar tal estudo – considerando o tempo de desenvolvimento do projeto – foi necessário delimitar o escopo com relação a: bases de dados, modelos de CNNs 2D pré-treinadas para a extração de características dos *frames* e técnicas de classificação (que serão usadas para classificar o tipo de lance).

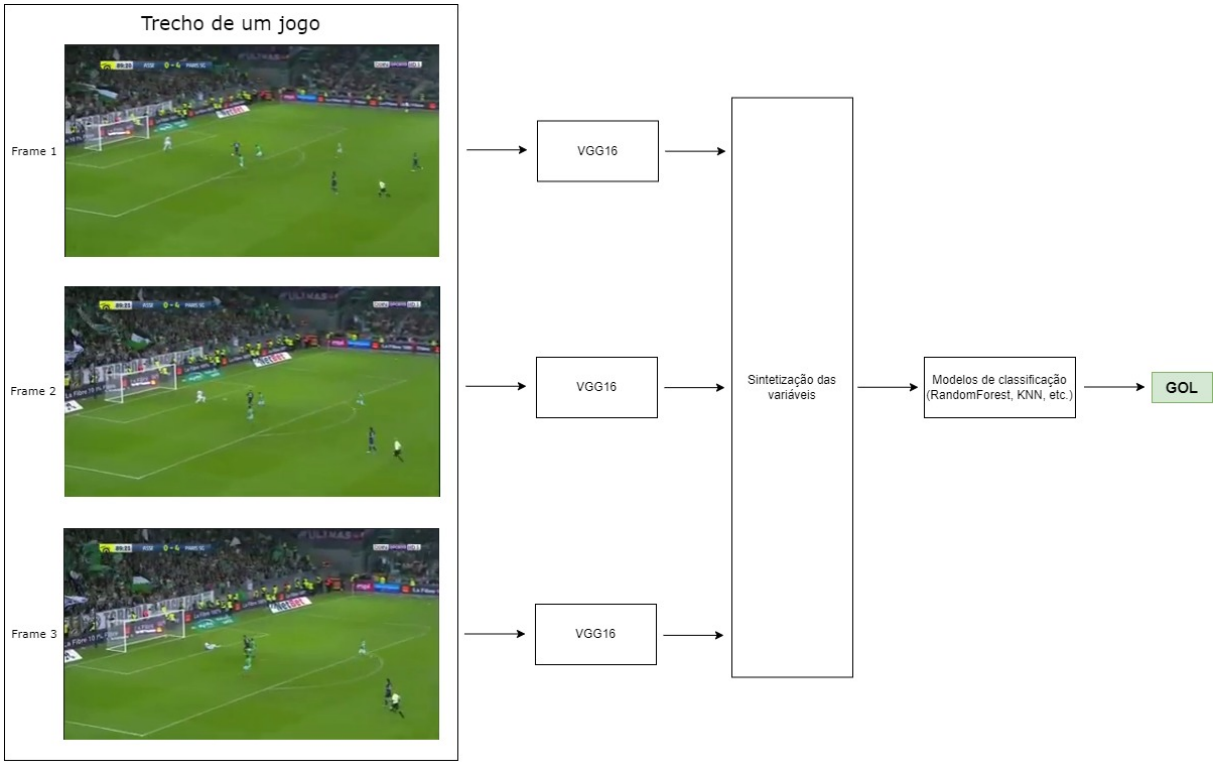
A primeira etapa desta definição se deu a partir de um estudo sobre as **bases de dados** encontradas em artigos acerca do tema. Estas bases são usadas em tarefas como detecção de objetos, detecção de melhores momentos (*highlights*) e classificação de *highlights*. Nesta etapa, optou-se por utilizar a base de dados *SoccerDB* (JIANG *et al.*, 2020). Esta base será explicada em maiores detalhes na Seção 3.3.2.

Em seguida, na segunda etapa, foram **extraídas as características de cada trecho (vídeo)** da base de dados *SoccerDB*, utilizando a CNN VGG16 pré-treinada na base de dados *ImageNet* (DENG *et al.*, 2009). Tal modelo foi projetado para fazer classificação de imagens, mas pode ser usado como extrator de características em cenários de *transfer learning*. Para adequar o uso da VGG16 ao contexto dos vídeos, foi necessário agregar os descritores de cada imagem (*frame*) de um vídeo para criar um descritor do vídeo único. Para agregar os descritores dos *frames* em um único descritor de video utilizou-se: média, mediana e desvio padrão. Esses processos estão ilustrados na figura 8 abaixo para melhor entendimento.

Por fim, na terceira etapa, foram analisados **modelos de classificação** para se usar neste contexto. Neste projeto, as abordagens escolhidas foram o *Random Forest* (CUTLER; CUTLER;

STEVENS, 2012) e o *LightGBM* (MICROSOFT, 2022). Tais técnicas foram utilizadas por tenderem a dar bons resultados e terem um custo computacional baixo quando comparado a outras técnicas (BENTÉJAC; CSÖRGŐ; MARTÍNEZ-MUÑOZ, 2021).

Figura 8 – Fluxo básico do projeto



Fonte: De autoria própria.

3.3 Descrição das Atividades Realizadas

Nessa seção, cada etapa citada anteriormente será detalhada, passo-a-passo, com o objetivo de evidenciar técnicas, ferramentas e recursos utilizados no projeto. Além disso, este detalhamento ajuda na reprodutibilidade deste projeto.

3.3.1 Definição da linguagem

A definição da linguagem se deu de forma clara e assertiva logo de início. A linguagem *Python* foi escolhida pois se adéqua muito bem a projetos envolvendo ciências de dados. Afinal, esta linguagem é versátil e possui diversas bibliotecas relacionados aos para manipulação de dados. A linguagem já havia sido trabalhada em projetos da disciplina de Introdução a Ciências de Dados, lecionada pelo professor orientador, o que facilitou o desenvolvimento dos códigos do projeto. A linguagem também possui grande comunidade engajada, boa documentação e é bem intuitiva.

Além disso, existe o suporte da ferramenta *Jupyter Notebook* à linguagem. Essa ferramenta quando usada na plataforma *Google Colab* disponibiliza um ambiente de execução completo, podendo se conectar com o *Google Drive*, plataforma de armazenamento em nuvem da *Google*, onde armazenaríamos o conteúdo das bases de dados utilizadas.

Finalmente, um fator que foi fundamental na escolha desse ambiente de desenvolvimento foi o fato do *Google Colab* disponibilizar GPU para aceleramento de *hardware* e alta memória. Tudo isso possibilitou a execução de processos que demandam muito poder computacional.

3.3.2 Estudo e manipulação das bases de dados

Desde o início do projeto, era sabido que seria necessário uma base de vídeos extensa e completa para criação do modelo de aprendizado. Após profunda pesquisa, a base *SoccerDB* foi escolhida por ser bem robusta e possuir boa documentação.

> Nós apresentamos uma desafiadora base de dados para uma compreensão abrangente de vídeos de futebol. Detecção de objetos, reconhecimento de ações, localização de ação temporal e detecção de melhores momentos. Essas ações podem ser investigadas de forma fechada em um ambiente restrito. (JIANG *et al.*, 2020, p. 3, tradução nossa)

Como citado, a base contempla conteúdo para diversos possíveis estudos. Para o reconhecimento de ações, substancial nesse projeto, a base engloba 346 vídeos de jogos de futebol completos. Desses, 270 vídeos são extraídos de outra base de dados, a *SoccerNet*. Os demais 76 vídeos são próprios da *SoccerDB*. Esses vídeos representam 142575 eventos, que individualmente pertencem a uma ou mais classes do catálogo.

Além dos vídeos, essa base inclui o catálogo de ações de cada lance de cada um dos 346 vídeos e também a separação entre lances que devem ser usados para treinamento e para teste dos modelos construídos.

A Tabela 1 detalha as classes dos eventos, além de dar um panorama quantitativo das mesmas na base.

Após a definição, foi necessário fazer o *download* dos vídeos de ambas as fontes - *SoccerDB* e *SoccerNet* - e *upload* no ambiente do *Google Drive*. Finalmente, a base de vídeos estava pronta para ser utilizada pelos *scripts* construídos no *Jupyter Notebook*.

3.3.3 Algoritmo de extração de características (VGG16)

O algoritmo para extração das variáveis que representam as imagens de cada evento foi elaborado com base no catálogo de ações disponibilizado pela *SoccerDB*. Esse arquivo indica um *id* para cada trecho (evento), o nome do vídeo relacionado a ele, os tempos de início e fim do evento, e finalmente, a classificação do evento - chute, gol, falta, etc. Assim, a ideia base do

Tabela 1 – Categorias dos eventos da base de dados

Código	Evento	Trechos na base	Proporção
0	Plano de fundo	121164	84,87%
1	Lesão	1327	0,93%
2	Cartão amarelo/vermelho	960	0,67%
3	Chute	11556	8,09%
4	Substituição	729	0,51%
5	Falta	2571	1,80%
6	Escanteio	2687	1,88%
7	Defesa	4278	2,99%
8	Cobrança de pênalti	128	0,09%
9	Cobrança de falta	4718	3,30%
10	Gol	2066	1,45%

Fonte: Jiang *et al.* ().

Nota: Como os eventos podem ser multi-classes, a soma da coluna de trechos não resulta no número total de evento da base. Analogamente, as porcentagens da última coluna não somam 100%

algoritmo é iterar sobre esse catálogo de ações disponibilizado e processar cada um dos trechos, gerando os descritores usando uma rede neural convolucional.

Como um mesmo vídeo contém dezenas de lances, optamos por copiar o vídeo do *Google Drive* para o ambiente de execução, para evitar possíveis gargalos de rede, já que seria necessário acessar cada vídeo diversas vezes. Dessa forma, para cada vídeo, verificamos se o vídeo já está no ambiente de execução e, caso não esteja, copiamo-lo para o ambiente local.

Para a leitura dos vídeos, foi utilizada uma biblioteca que permite acesso a *frames* não sequenciais de forma mais rápida: a biblioteca *decord* (COMMUNITY, 2022). Ao fazer a abertura do vídeo, também já é realizado o cálculo da duração do vídeo, utilizando um comando *ffprobe* (TOMAR, 2006), que será útil para encontrar o index dos *frames* de cada trecho.

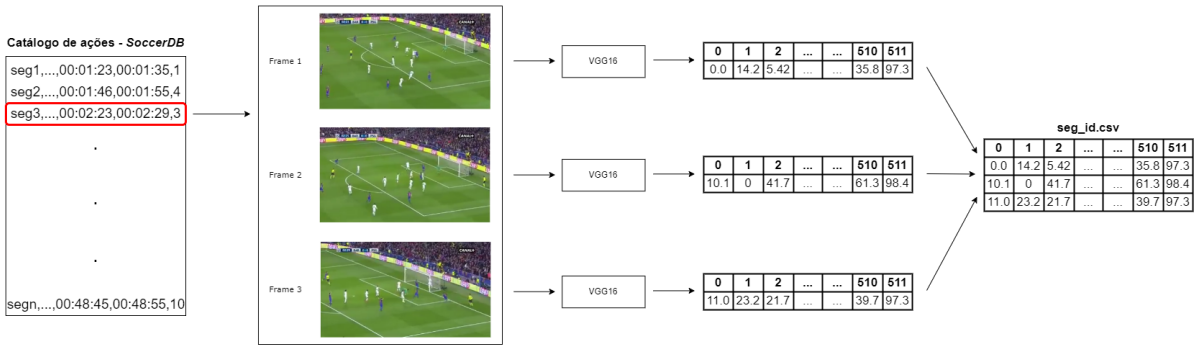
Após a abertura dos vídeos, é executada a função de obter os *frames* do trecho que está sendo processado. Para isso, são necessários: o cursor do vídeo que foi aberto anteriormente; a duração do vídeo; os tempos de início e fim do evento; e o intervalo de *frames* utilizado. Esse intervalo indica quantos *frames* do vídeo serão ignorados após cada *frame* extraído para gerar os descritores (variáveis), haja vista que gerar os descritores para cada um dos *frames* seria computacionalmente muito custoso. Para esse estudo, foi utilizado um intervalo de 5 *frames*. Ou seja, a cada 5 *frames* do trecho do evento, uma imagem é capturada para posteriormente ser processada.

Obtidos os *frames* do evento, o próximo passo é gerar os descritores para cada um deles. Para isso, foi escolhida a VGG16 (TensorFlow/Keras) (CHOLLET *et al.*, 2015), pré-treinada a partir da base *ImageNet*. O principal detalhe é remover a parte densa (*top*) da VGG16, que realizaria a predição (classificação) de fato, pois a ideia é apenas gerar os descritores e depois, usando *transfer learning*, aplicar os resultados em outro modelo de classificação. Além disso, para o *pooling*, foi escolhido o 'máximo' para a redução das dimensões de saída. Assim, teremos

uma saída de 512 variáveis para cada *frame*.

Então, os *frames* do evento são redimensionados para 224×224 pixels, correspondente ao padrão de entrada da VGG16, e depois são pré-processados - canais RGB são reordenados, escalas de pixels são redimensionadas, etc. Finalmente, são gerados os 512 descritores de cada *frame* do evento. Na Figura 9, está ilustrado o fluxo básico do algoritmo para contribuir com a compreensão.

Figura 9 – Ilustração do algoritmo de geração dos descritores



Fonte: De autoria própria.

Os descritores de todas as imagens que compõem um evento são dispostos em um *DataFrame* e então armazenados em um arquivo `.csv`, único para cada evento. Esse processo se repete para cada um dos 142575 eventos catalogados.

Ao final da execução do algoritmo de geração dos descritores, temos então um arquivo `.csv` para cada um dos 142575 eventos dos jogos de futebol. Cada um dos arquivos armazena os 512 descritores de cada um dos *frames* selecionados daquele evento. A quantidade de *frames* por evento é variável, seguindo o intervalo de 5 *frames* citado.

3.3.4 Algoritmo de sintetização das características

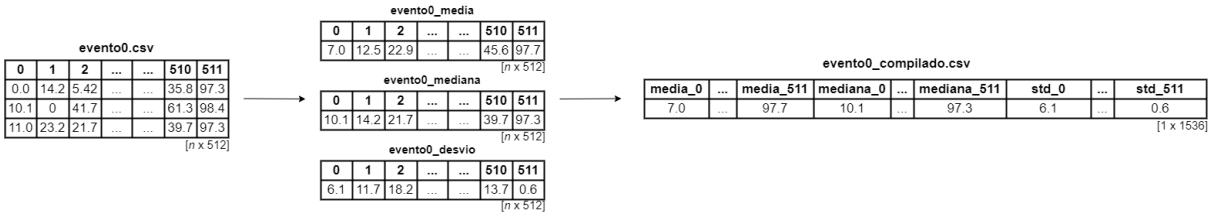
Após a descrição das imagens que compõem os eventos, os arquivos `.csv` de saída seguiam o formato $n \times 512$, onde n representa o número de *frames* capturados daquele evento. O objetivo do algoritmo de sintetização é compilar esses valores do `.csv` e gerar um vetor único de saída para cada evento, que servirá de entrada para o modelo de classificação utilizado em seguida.

Para sintetizar as características, diversas estratégias poderiam ser utilizadas. Essa etapa é muito relevante para o resultado final do modelo de classificação. Como tratamos cada *frame* como independente na extração das características, nessa etapa o objetivo é tentar criar uma relação entre os *frames*, mesmo que de forma superficial. A abordagem escolhida foi compilá-los utilizando os três principais métodos de estatística descritiva: média, mediana e desvio padrão. Ao

final do algoritmo, teremos um vetor para cada evento da base, todos em um mesmo *dataframe* - arquivo `.csv`, após exportado.

Assim, para cada evento, calculamos a média, a mediana e o desvio padrão de cada uma das 512 variáveis dos *frames* computados pela VGG16. Logo, ao final da execução, cada trecho será representado por 1536 variáveis, dispostas em um tensor 1×1536. A Figura 10 mostra a saída da sintetização dos descritores para um evento representado por 3 *frames*.

Figura 10 – Exemplo do algoritmo de sintetização das características.



Fonte: De autoria própria.

O *output* desse algoritmo representa diretamente o *input* do modelo de classificação detalhado a seguir.

3.3.5 Algoritmo de classificação

O algoritmo de classificação é o responsável por aprender com os eventos de treino e ser capaz de classificar com qualidade novos eventos (lances) do jogo de futebol. Foram escolhidos duas técnicas de *ensembles* para a realização de experimentos sobre tal tarefa: o *Random Forests* e o *LightGBM* (uma variação do *Gradient Boosting*).

O *dataset* escolhido (*SoccerDB* (JIANG *et al.*, 2020)) já possui uma sugestão de separação entre dados de treinamento e de teste, esse processo foi simples nessa etapa de classificação. Inicialmente, foi lido o arquivo `.csv` exportado do algoritmo de sintetização de características, contendo médias, medianas e desvios padrão. Em seguida, a separação foi realizada conforme sugerido pelo autor do *dataset*.

Com os dados de treino e teste bem definidos, o próximo passo foi o treinamento dos modelos de classificação. Os *ensembles* de classificação possuem hiperparâmetros que podem ser configurados, e assim contribuir para o modelo final. Para encontrar os melhores hiperparâmetros de cada modelo, foi feito um *grid-search* dividindo-se a base de treinamento em duas partes (60% dos dados para treinamento e 40% para validação). Para executar esse processo foi usado a classe *GridSearchCV*, disponibilizada pela biblioteca *sklearn* (PEDREGOSA *et al.*, 2011b). Tal ferramenta automatiza o processo de combinação dos hiperparâmetros de treinamento de um modelo e determina a melhor combinação encontrada.

Como o problema permite que um evento pertença a mais de uma classe (*multi-output*), foi necessário utilizar o *MultiOutputClassifier* (PEDREGOSA *et al.*, 2011a) em conjunto com os

algoritmos de classificação. Essa função transforma o problema em uma classificação binária para cada classe, indicando a existência ou não da mesma na instância.

As métricas escolhidas para avaliar os modelos foram: o *mAP* (*Mean Average Precision*) e a área sob a curva ROC (ROC AUC). Os modelos *RandomForest* e *LightGBM* de melhores performances serão os treinados e avaliados na base de teste, posteriormente.

Para o classificador *RandomForest*, foram considerados os hiperparâmetros de configuração: número de árvores de decisão e máxima profundidade da árvore. Os valores utilizados no *GridSearchCV* para esses hiperparâmetros do *RandomForest* estão mostrados na Tabela 2.

Tabela 2 – Valores considerados pelo *GridSearchCV* para o *RandomForest*.

Hiperparâmetro	Valores testados
Número de árvores de decisão	[10, 50, 100, 500, 1000]
Máxima profundidade da árvore	[2, 3, 5, 10]

Para o *LightGBM*, foram experimentados diferentes valores para: a taxa de aprendizado do modelo, número de árvores de decisão, profundidade máxima da árvore. Os valores considerados pelo *GridSearchCV* nesse caso estão listados na Tabela 3.

Tabela 3 – Valores considerados pelo *GridSearchCV* para o *LightGBM*.

Hiperparâmetro	Valores testados
Taxa de aprendizado	[0.05, 0.1]
Número de árvores de decisão	[10, 50, 100, 500, 1000]
Máxima profundidade da árvore	[2, 3, 5, 10]

Após o treinamento dos diversos modelos utilizando ambas as técnicas, foi selecionado o melhor modelo de cada um dos *ensembles* treinados. Em seguida, esses modelos foram novamente treinados, mas agora utilizando toda a base de treino disponível. Finalmente, eles foram avaliados na base de eventos de teste determinada previamente. Os resultados serão descritos a seguir.

3.4 Resultados Obtidos

A partir das execuções do *GridSearchCV* para os modelos do *RandomForest* e do *LightGBM*, foi possível encontrar os hiperparâmetros de configuração que apresentaram melhores performances (*scores*). Para isso, foi utilizada a métrica *mAP*. O modelo do *RandomForest* que apresentou melhor resultado na base de treinamento foi construído a partir dos hiperparâmetros listados na Tabela 4. A precisão média de acertos obtida para esse modelo foi de $32,0\%$.

Para o *LightGBM*, os hiperparâmetros que apresentaram o melhor resultado foram os listados na Tabela 5. Nesse caso, o *mAP* foi de $43,7\%$.

Tabela 4 – *RandomForest* de melhor performance na base de treinamento.

Hiperparâmetro	Valor
Número de árvores de decisão	1000
Máxima profundidade da árvore	10

Tabela 5 – *LightGBM* de melhor performance na base de treinamento.

Hiperparâmetro	Valor
Taxa de aprendizado	0.1
Número de árvores de decisão	1000
Máxima profundidade da árvore	10

Os modelos selecionados foram então treinados utilizando toda a base de treino. Em seguida, foi realizada a predição das classes dos eventos da base de teste. As avaliações finais dos modelos foram dadas conforme as métricas ROC AUC e precisão média. Os resultados estão detalhados e discutidos a seguir.

3.4.1 Precisão média

Na Tabela 6, estão listados os *scores* de cada classe, além da média das precisões das mesmas (*mAP - Mean Average Precision*), para cada um dos modelos.

Tabela 6 – Precisão por classe e média dos modelos de classificação treinados.

Evento	***RandomForest***	***LightGBM***
P. de fundo	96,3%	97,7%
Lesão	23,7%	32,6%
Cartão	17,6%	40,0%
Chute	47,3%	69,3%
Substituição	52,2%	85,3%
Falta	17,7%	31,9%
Escanteio	40,4%	70,3%
Defesa	16,7%	22,6%
Cob. pênalti	1,4%	5,0%
Cob. falta	17,6%	26,4%
Gol	21,2%	30,9%
Média	**32,0%**	**46,6%**

A efeito de comparação, a Tabela 7 traz os resultados do trabalho Jiang *et al.* (2020), que também buscou classificar os lances do futebol. O autor utiliza a mesma base de dados e a mesma metodologia de teste utilizadas nesse projeto, mas traz uma abordagem mais complexa. São utilizadas redes neurais voltadas para vídeos, como a *PySlowFast* (SF, na tabela) e a *Non-Local* (NL, na tabela). Essas redes neurais já recebem como entrada uma sequência de *frames* e são capazes de interpretar a relação entre eles, diferentemente da abordagem de média, mediana e desvio padrão adotada nesse projeto. Além disso, em paralelo, o autor também desenvolveu

Tabela 7 – Precisão média obtida para cada classe em outro projeto da área.

Evento	SF-32	NL-32	SF-64	NL-64	MRTS
P. de Fundo	99,08%	99,16%	99,32%	99,26%	99,44%
Lesão	23,03%	36,06%	22,56%	37,70%	39,14%
Cartão	28,62%	36,74%	46,62%	48,83%	60,64%
Chute	82,98%	85,32%	88,25%	85,17%	90,19%
Substituição	92,34%	90,60%	93,44%	90,30%	92,24%
Falta	73,33%	72,92%	77,34%	74,30%	73,46%
Escanteio	91,76%	91,82%	93,16%	91,92%	92,62%
Defesa	38,91%	40,77%	52,24%	42,17%	52,19%
Cob. pênalti	63,02%	48,51%	73,48%	53,36%	67,00%
Cob. falta	64,75%	65,75%	67,78%	68,01%	70,09%
Gol	31,89%	31,92%	47,44%	39,94%	56,23%
Média	**62,70%**	**63,60%**	**69,24%**	**66,45%**	**72,11%**

Fonte: Jiang *et al.* (2020).

um modelo de detecção de objetos para atuar em conjunto com o reconhecimento de ações e potencializar a classificação dos eventos.

A última linha da Tabela 7 mostra os valores de *mAP* obtidos para os diferentes métodos de classificação da pesquisa. Nota-se que os resultados ficaram na margem de 60% a 70%. Quando comparamos com o melhor resultado obtido aqui nesse trabalho, uma precisão média de $46,6\%$ do modelo *LightGBM*, concluímos que a abordagem mais simples aqui adotada se mostra interessante, visto que com técnicas com menor uso de recursos computacionais, foi possível obter um bom *score* se comparado aos trabalhos mais robustos da área.

3.4.2 ROC AUC

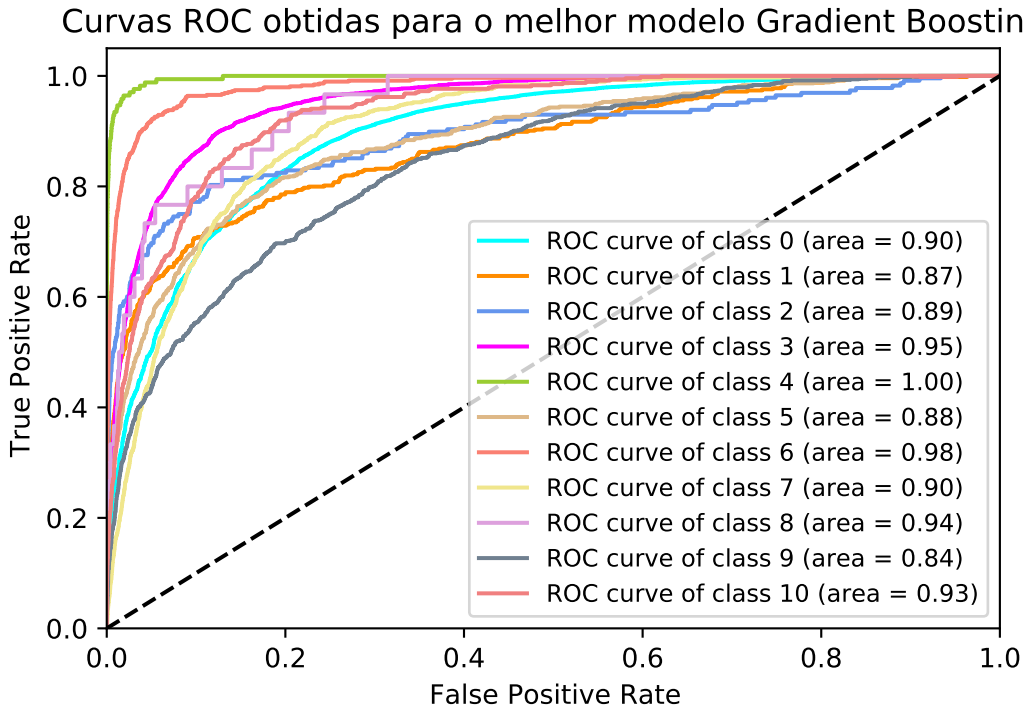
Na revisão da literatura notou-se que o *mAP* não é a única métrica utilizada no estudo dos resultados da classificação de lances. Outra abordagem, utilizada em estudos como o de Sen *et al.* (2022), é o cálculo da AUC média das classes e a apresentação da curva ROC de cada uma das classes. Sendo assim, na Tabela 8, estão mostrados os *scores* obtidos para cada classe possível, além do *score* médio de todas as classes, para cada um dos modelos, conforme a ROC AUC. Além disso, na Figura 11, estão mostradas as curvas ROC resultantes do modelo *LightGBM*, para cada classe.

É importante observar que – embora tenha-se calculado as mesmas métricas de Sen *et al.* (2022) – não é possível fazer uma comparação direta entre as duas abordagens. Isso acontece porque as bases de dados utilizadas nos experimentos são diferentes. Ainda assim, os resultados mostrados na Tabela 8 e na Figura 11 indicam que a abordagem proposta foi capaz de distinguir diferentes tipos de lance, o que faz desse projeto um bom ponto de partida no estudo da classificação dos eventos do futebol.

Tabela 8 – *Scores* ROC AUC por classe e médio dos modelos de classificação treinados.

Evento	***RandomForest***	***LightGBM***
P. de fundo	83,7%	89,7%
Lesão	82,4%	87,3%
Cartão	90,5%	89,3%
Chute	91,1%	95,1%
Substituição	98,3%	99,7%
Falta	85,4%	88,4%
Escanteio	95,9%	98,1%
Defesa	87,4%	90,3%
Cob. pênalti	93,6%	94,4%
Cob. falta	81,7%	84,2%
Gol	90,8%	93,3%
Média	**89,2%**	**91,8%**

Figura 11 – Curvas ROC obtidas para o modelo *LightGBM*.



Fonte: De autoria própria.

3.5 Dificuldades e Limitações

Diversas foram as dificuldades e os imprevistos encontrados durante o projeto. Porém, nenhum representou um impedimento muito grande a ponto de bloquear ou colocar o projeto em risco. Os primeiros empecilhos encontrados foram relacionados às bases de dados. A utilização da mesma envolveu trocas de *emails* com os donos das bases, que são em sua maioria chineses. Além do fator relacionado a linguagem e comunicação, a base era disponibilizada em um serviço na nuvem restrito a usuários chineses, o que fez necessário um profundo estudo sobre como obter

os arquivos da base, para então fazer o *upload* dos mesmos no *Google Drive*, o que facilitaria muito o desenvolvimento do projeto nas etapas seguintes.

Em seguida, na etapa de obtenção dos *frames* dos eventos, alguns problemas também surgiram e exigiram novas soluções. O primeiro deles foi em relação a leitura dos vídeos da base. Foi necessário encontrar uma biblioteca que tornasse possível o acesso a um *frame* de qualquer ponto do vídeo de forma eficiente, o que se mostrou bastante desafiador. Com a biblioteca ideal encontrada, a ideia inicial era gerar diversos vídeos a partir dos eventos e armazená-los para depois processá-los. Após diversos testes, essa estratégia se mostrou ineficiente, ocupando muito tempo e também exigindo mais espaço de armazenamento. A solução encontrada foi ler os vídeos e trabalhar com ele em memória, durante o tempo de execução do *script*.

A geração dos descritores foi uma etapa que exigiu paciência e estudo sobre como aumentar a velocidade de processamento dos dados, visto que a base conta com mais de 142000 eventos (trechos de vídeos), com vários *frames* em cada um deles. Nesse ponto, o professor forneceu um recurso que foi essencial para a conclusão dessa etapa: o ambiente mais robusto do *Google Collaboratory*, com GPU e mais memória RAM. Esse recurso também contribuiu muito nas fases seguinte, como compilação dos descritores e criação dos modelos de classificação, que não apresentaram grandes dificuldades.

3.6 Considerações Finais

Nesse capítulo, foi descrito com detalhes como se deu o desenvolvimento do projeto, além de seus resultados e dificuldades encontradas. Os algoritmos foram explicados e ilustrados, acompanhando exemplos e informações sobre o processo de execução. Os resultados foram tabelados e analisados, comparando os modelos de classificação treinados e suas capacidades preditivas. A seguir, serão apresentadas as conclusões do projeto.

Capítulo 4

CONCLUSÃO

4.1 Contribuições

O projeto propõe uma outra abordagem para classificação de eventos relacionados ao futebol em relação à maioria dos trabalhos já existentes na área, utilizando uma estratégia mais simples para interpretação dos vídeos dos lances. A utilização das redes neurais convolucionais para os vídeos, acoplada aos *ensembles* classificadores, pode representar uma perspectiva diferente nessa área de estudo.

Em relação aos resultados obtidos, é possível concluir que foram satisfatórios, visto que os modelos classificadores foram capazes de adquirir boa capacidade preditiva, dentro da base de vídeos proposta. Ainda que os modelos não apresentem *scores* tão elevados quanto os descritos em outros trabalhos na literatura, esse projeto representa de fato uma abordagem mais simples de ser construída, que não exige tantos recursos e também tempo, de forma que se adequasse aos objetivos deste trabalho.

Pessoalmente, considero que todo o processo de estudo e desenvolvimento envolvido nesse trabalho representou um grande aprendizado. Foi extremamente interessante aliar os interesses acadêmicos com os pessoais e conseguir obter mais conhecimento sobre esses assuntos. Estudar os trabalhos já existentes na área, conseguir imaginar os projetos implementados e desenvolver algo no mesmo sentido fez esse trabalho muito especial.

Além disso, também valorizo o conhecimento que é inerente ao processo. Ao esbarrar nos problemas e erros relacionados ao desenvolvimento, a busca por soluções sempre levou a um novo aprendizado. Como exemplo, cito as diversas bibliotecas pesquisadas para que a manipulação dos vídeos fosse realizada de forma satisfatória e também as noções de otimização que o professor orientador sempre buscou passar para que tornássemos o projeto mais eficiente.

4.2 Considerações Sobre o Curso de Graduação

Considero essenciais as disciplinas básicas de programação e lógica, ministradas no início da minha graduação. Entendo que matérias como Introdução à Ciência da Computação e Estruturas de Dados foram primordiais para o meu desenvolvimento acadêmico e também para

este projeto. Apresentaram obstáculos no início, porém vejo que os primeiros passos são sempre os mais difíceis, mas são também os mais didáticos.

Além disso, as disciplinas de Visão Computacional e Introdução às Ciências de Dados - esta última ministrada com maestria pelo professor orientador deste trabalho - foram fundamentais para esclarecer os meus objetivos para o projeto de formatura e também me aprofundar nos assuntos de meu interesse. Infelizmente, não tive a oportunidade de cursar a disciplina de Redes Neurais, mas, com esse projeto e com a orientação do professor Tiago, consegui aprender bastante sobre o assunto.

Em relação ao curso, vejo que a Engenharia de Computação muitas vezes se mostra carregada em alguns momentos da graduação. Espaços livres na grade horária poderiam levar o aluno a se desenvolver em áreas específicas de seu interesse, através de disciplinas optativas ou extra-curriculares. No curso, muitas vezes acontece uma sobrecarga e é difícil conciliar as atividades que envolvem a universidade, como um todo.

Apesar disso, as minhas áreas de interesse foram abordadas, em geral, ao longo da graduação. Em momento algum, arrependi-me ou pensei sobre uma possível mudança de curso. A universidade fornece uma ótima infra-estrutura que atende perfeitamente ao curso de Engenharia de Computação. Em relação aos docentes, o nível é ótimo, tratando-se de conhecimento. A didática nem sempre é um destaque nos professores, e isso muitas vezes afeta o aprendizado, mas na maioria das vezes a quantidade de recursos disponíveis para dar suporte ao aluno acaba compensando tal defasagem.

4.3 Trabalhos Futuros

Como trabalho futuro, propõe-se o aprofundamento dos estudos da aplicação de redes neurais convolucionais pré-treinadas no âmbito do futebol. A utilização de outras redes neurais convolucionais na extração de características - DenseNet (HUANG *et al.*, 2016), ResNet (HE *et al.*, 2015), etc. - seria uma opção para a continuidade do estudo nessa área, a fim de comparar resultados e entender possíveis diferenças.

Além disso, propõe-se a utilização de outros algoritmo de classificação para determinação das classes dos eventos. Nesse trabalho, foram escolhidos os *ensembles* de classificação *RandomForest* e *Gradient Boosting*, mas seria interessante também analisar os resultados em outros modelos.

Finalmente, o desenvolvimento de um modelo de classificação próprio para os vídeos dos eventos também é proposto como trabalho futuro. No projeto, eram feitas as extrações e processamentos dos *frames* dos vídeos, para então classificá-los. Em um outro trabalho, poderia ser abordada uma rede neural que já trabalha com vídeos, o que certamente resultaria em modelos únicos para a classificação dos eventos do futebol.

REFERÊNCIAS

BENTÉJAC, C.; CSÖRGŐ, A.; MARTÍNEZ-MUÑOZ, G. A comparative analysis of gradient boosting algorithms. **Artificial Intelligence Review**, v. 54, n. 3, p. 1937–1967, Mar 2021. ISSN 1573-7462. Disponível em: <https://doi.org/10.1007/s10462-020-09896-5>. Citado na página 25.

BONFIM, K. **Como são feitas as imagens aéreas da copa**. 2018. Disponível em: <https://medium.com/droneiro-com/como-s%C3%A3o-feitas-as-imagens-a%C3%A9reas-da-copa-fd27cf75534f>. Citado na página 12.

BRAMER, M. Ensemble classification. In: ____. **Principles of Data Mining**. London: Springer London, 2013. p. 209–220. ISBN 978-1-4471-4884-5. Disponível em: <https://doi.org/10.1007/978-1-4471-4884-5_14>. Citado na página 18.

CAMACHO, K. **6 maneiras que a tecnologia está transformando o esporte**. 2022. Disponível em: <https://revistacapitaleconomico.com.br/tecnologia-no-esporte/>. Citado na página 12.

CHOLLET, F. *et al.* **Keras**. GitHub, 2015. Disponível em: <https://github.com/fchollet/keras>. Citado na página 27.

COMMUNITY, D. D. M. L. **Decord**. 2022. Disponível em: <https://github.com/dmlc/decord>. Citado na página 27.

CUTLER, A.; CUTLER, D. R.; STEVENS, J. R. Random forests. In: ____. **Ensemble Machine Learning: Methods and Applications**. Boston, MA: Springer US, 2012. p. 157–175. ISBN 978-1-4419-9326-7. Disponível em: <https://doi.org/10.1007/978-1-4419-9326-7_5>. Citado na página 25.

DENG, J.; DONG, W.; SOCHER, R.; LI, L.-J.; LI, K.; FEI-FEI, L. Imagenet: A large-scale hierarchical image database. In: **2009 IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2009. p. 248–255. Citado 2 vezes nas páginas 20 e 24.

ESPORTIVA, G. **Como a tecnologia no futebol auxilia atletas e clubes**. 2021. Disponível em: <https://www.gazetaesportiva.com/institucional/como-a-tecnologia-no-futebol-auxilia-atletas-e-clubes/>. Citado na página 12.

FAKHAR, B.; KANAN, H. R.; BEHRAD, A. Event detection in soccer videos using unsupervised learning of spatio-temporal features based on pooled spatial pyramid model. **Multimedia Tools and Applications**, v. 78, n. 12, p. 16995–17025, Jun 2019. ISSN 1573-7721. Disponível em: <https://doi.org/10.1007/s11042-018-7083-1>. Citado na página 13.

FIFA. **Video Assistant Referee (VAR) Technology**. 2022. Disponível em: <https://www.fifa.com/technical/football-technology/standards/video-assistant-referee>. Citado na página 12.

G, R. **Everything you need to know about VGG16**. 2021. Disponível em: <https://medium.com/@mygreatlearning/everything-you-need-to-know-about-vgg16-7315defb5918>. Citado na página 16.

HE, K.; ZHANG, X.; REN, S.; SUN, J. **Deep Residual Learning for Image Recognition**. arXiv, 2015. Disponível em: <https://arxiv.org/abs/1512.03385>. Citado na página 36.

HUANG, G.; LIU, Z.; MAATEN, L. van der; WEINBERGER, K. Q. **Densely Connected Convolutional Networks**. arXiv, 2016. Disponível em: <https://arxiv.org/abs/1608.06993>. Citado na página 36.

IBM. **Convolutional Neural Networks**. 2020. Disponível em: <https://www.ibm.com/cloud/learn/convolutional-neural-networks>. Citado 3 vezes nas páginas 14, 15 e 16.

JIANG, Y.; CUI, K.; CHEN, L.; WANG, C.; XU, C. **SoccerDB: A Large-Scale Database for Comprehensive Video Understanding**. Disponível em: <https://github.com/newsdata/SoccerDB>. Citado na página 27.

____. SoccerDB: A large-scale database for comprehensive video understanding. ACM, oct 2020. Disponível em: <https://doi.org/10.1145%2F3422844.3423051>. Citado 5 vezes nas páginas 24, 26, 29, 31 e 32.

KARIMI, A.; TOOSI, R.; AKHAEE, M. A. Soccer event detection using deep learning. **CoRR**, abs/2102.04331, 2021. Disponível em: <https://arxiv.org/abs/2102.04331>. Citado na página 21.

MICROSOFT. **LightGBM**. 2022. Disponível em: <https://lightgbm.readthedocs.io/_/downloads/en/latest/pdf/>. Citado na página 25.

MONGODB. **Unstructured Data**. 2022. Disponível em: <https://www.mongodb.com/unstructured-data>. Citado na página 14.

NOONEY, K. **Deep dive into multi-label classification..! (With detailed Case Study)**. 2018. Disponível em: <https://towardsdatascience.com/journey-to-the-center-of-multi-label-classification-384c40229bff>. Citado na página 17.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, E. **MultiOutputClassifier**. 2011. Disponível em: <https://scikit-learn.org/stable/modules/generated/sklearn.multioutput.MultiOutputClassifier.html>. Citado na página 29.

____. Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011. Citado na página 29.

SAHA, S. **A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way**. 2018. Disponível em: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>. Citado na página 16.

SEN, A.; HOSSAIN, S. M. M.; RUSSO, M. A.; DEB, K.; JO, K.-H. Fine-grained soccer actions classification using deep neural network. In: **2022 15th International Conference on Human System Interaction (HSI)**. [S.l.: s.n.], 2022. p. 1–6. Citado 3 vezes nas páginas 13, 22 e 32.

SINGH, A.; THAKUR, N.; SHARMA, A. A review of supervised machine learning algorithms. In: **2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)**. [S.l.: s.n.], 2016. p. 1310–1315. Citado na página 17.

STEEN, D. **Understanding the ROC Curve and AUC**. 2020. Disponível em: <https://towardsdatascience.com/understanding-the-roc-curve-and-auc-dd4f9a192ecb>. Citado na página 19.

TOMAR, S. Converting video formats with ffmpeg. **Linux Journal**, Belltown Media, v. 2006, n. 146, p. 10, 2006. Citado na página 27.

GLOSSÁRIO

Frame: ou quadro de vídeo. É cada uma das imagens que compõem um vídeo.