Revista Brasileira de Computação Aplicada, Novembro, 2023

DOI: 10.5335/rbca.v15i3.14743
Vol. 15, Nº 3, pp. 1–14
Homepage: seer.upf.br/index.php/rbca/index

ARTIGO ORIGINAL

Categorização de ações em vídeos de futebol utilizando uma arquitetura CNN-RNN

Categorization of actions in soccer videos using a CNN-RNN architecture

Matheus de Sousa Macedo[1], and Diana Francisca Adamatti[1]

[1]Centro de Ciências Computacionais –Universidade Federal do Rio Grande - C3/FURG

*{macedo.matheus81, dianaadamatti}@furg.br

Recebido: 12/04/2023. **Revisado:** 26/10/2023. **Aceito:** 16/11/2023.

Resumo

A extração de informações semânticas de vídeos de futebol tem diversas aplicações, como publicidade contextual, resumo de partidas e extração de destaques. As aplicações de análise de vídeos de futebol podem ser categorizadas em Detecção de Ações, Rastreamento de jogadores e/ou bola e Análise de jogo. Utiliza-se como base de dados uma versão modificada do *Dataset* SoccerNet-v2, afim de reduzir o Poder Computacional mínimo exigido. A tarefa de Detecção de Ações torna-se difícil por conta da sobreposição de ações e também por causa das condições de captura de vídeo que tem diversos ângulos, anúncios e cortes de câmera. Para superar esses desafios, a Rede Neural Convolucional (CNN) e a Rede Neural Recorrente (RNN) são utilizadas em conjunto para classificar diferentes comprimentos de vídeos de ações do futebol. Utiliza-se uma CNN, InceptionV3, pré-treinada para a extração de características espaciais. Posteriormente, uma RNN, Unidades Recorrentes Fechadas (GRU), para o reconhecimento de sequências, que trata a dependência temporal e resolve o problema do desaparecimento de gradiente. Por fim, a camada *Softmax* atribui probabilidades decimais a cada classe. Chega-se a uma configuração de rede, com quatro ações classificáveis, e uma acurácia de 94%.

Palavras-Chave: Ações de Futebol; Classificação de Vídeos; Redes Neurais Convolucionais; Redes Neurais Recorrentes

Abstract

The extraction of semantic information from soccer videos has several applications, such as contextual advertising, match summaries, and highlight extraction. Applications for analyzing soccer videos can be categorized into Action Detection, Player and/or Ball Tracking, and Game Analysis. A modified version of the SoccerNet-v2 Dataset is used as a database to reduce the minimum computational power required. The task of Action Detection becomes challenging due to the overlap of actions and the various video capture conditions that include multiple angles, ads, and camera cuts. To overcome these challenges, a Convolutional Neural Network (CNN) and a Recurrent Neural Network (RNN) are used together to classify different lengths of soccer action videos. A pre-trained CNN, InceptionV3, is used for spatial feature extraction, and a Gated Recurrent Unit (GRU) RNN is used for sequence recognition, which addresses temporal dependence and solves the problem of gradient disappearance. Finally, the Softmax layer assigns decimal probabilities to each class. A network configuration with four classifiable actions and an accuracy of 94% is achieved.

Keywords: Convolutional Neural Networks; Recurrent Neural Networks; Soccer Actions; Video Classification.

1 Introdução

Aprendizado de Máquina (*Machine Learning*) é a utilização de algoritmos para extrair informações de dados brutos e representá-los através de algum modelo matemático. Esse modelo, é utilizado para fazer inferências em outros conjuntos de dados (Szeliski, 2010).

Uma das técnicas utilizadas atualmente em tarefas de Aprendizado de Máquinas são as Redes Neurais Convolucionais, que empregam operações de convolução: são operações matemáticas em duas funções que produzem uma terceira função que expressa como a forma de uma é modificada pela outra, ao invés da multiplicação de matrizes simples aplicadas em redes neurais tradicionais (Ioannidou et al., 2017).

Uma área que utiliza o Aprendizado de Máquina é a Visão Computacional, que atrai pesquisadores do mundo todo para a classificação de ações em vídeos, pois embora muitas pesquisas sejam conduzidas nesta área, ainda temos problemas desafiadores tanto em vídeos gravados como em tempo real. Desta maneira, muitos trabalhos estão focados em como extrair informações semânticas destes vídeos (Baccouche et al., 2010).

A área dos esportes está se tornando popular entre os pesquisadores Baccouche et al. (2010), porém mesmo o futebol sendo um dos esportes mais populares e assistidos no mundo, ainda existem poucos trabalhos com este embasamento (que será mostrado durante a Revisão Sistemática da Literatura). A ideia é **classificar as ações do futebol e categorizá-las**, com o objetivo de eliminar a tarefa manual e tediosa de geração de destaques, cálculos estatísticos e resumo das partidas, utilizando dois tipos de Redes Neurais em conjunto, uma Rede Neural Convolucional (CNN) juntamente com uma Rede Neural Recorrente (RNN). É uma tarefa desafiadora, devido a sobreposição de cenas entre as diferentes ações, com isso, infelizmente muitas ações principais são ignoradas. Essa aplicação tem alto impacto comercial nas emissoras (Sen and Deb, 2022).

Como base de dados, utiliza-se o Dataset Anotado SoccerNet-v2 (Deliege et al., 2021). Porém, como é exigido um poder computacional muito alto para utilizar a sua estrutura padrão, a mesma foi reestruturada, como apresentado na Seção 3.1.2.

2 Embasamento teórico e aplicado

2.1 Deep Learning

As redes neurais artificiais já existiam desde a década de 1950, porém ainda faltavam recursos para que os modelos realmente funcionassem de forma rápida, com uma grande quantidade de dados (*Big Data*). Com o avanço em unidades de processamento, por exemplo GPUs, a execução tornou-se extremamente rápida, sendo possível a execução de Redes Neurais com diversos níveis. Os modelos de Deep Learning são exemplos deste tipo de arquitetura (Ioannidou et al., 2017).

No início dos anos 2000, o poder computacional expandiu exponencialmente e o mercado viu um crescimento de técnicas computacionais que não eram possíveis antes disso. Foi quando o aprendizado profundo (*Deep Learning*) emergiu do crescimento computacional explosivo dessa década como o principal mecanismo de construção de sistemas de Inteligência Artificial, ganhando muitas competições importantes de aprendizagem de máquina. O interesse por *Deep Learning* não para de crescer e hoje vemos o termo aprendizado profundo sendo mencionado com frequência cada vez maior e soluções comerciais surgindo a todo momento. E uma das maiores aplicações do *Deep Learning* é a Visão Computacional (Szeliski, 2010).

2.2 Visão Computacional

Os seres humanos tem a capacidade de visualizar a estrutura tridimensional do mundo com certa facilidade. Distinguir cores, sombreamento, profundidade e padrões de luz. Também são capazes de olhar uma fotografia e contar a quantidade de indivíduos e até nomeá-los, distinguir rostos e também perceber que emoções aqueles rostos transmitem, o que é vivo e o que não é. A psicologia e a medicina levaram décadas para descobrirem como a visão funciona e ainda assim há alguns quebra-cabeças desta área não solucionados (Szeliski, 2010).

Com isso, pesquisadores de Visão Computacional vêm desenvolvendo, de forma paralela, soluções matemáticas para modelar corpos tridimensionais e detectar objetos em imagens e vídeos (Szeliski, 2010). Com estes avanços, já é possível visualizar um pouco das situações cotidianas do nosso dia-a-dia, por exemplo, uma reconstrução de um ambiente 3D através de milhares de imagens sobrepostas, quando utilizamos o Google Street View, ou a sugestão de rostos utilizando o Google Fotos. Também conseguimos visualizar a detecção de rosto ao utilizá-lo no desbloqueio de tela, e por fim a detecção de objetos utilizando o Google Lens.

As técnicas de *Deep Learning* mais utilizadas na Visão Computacional são as:

- Redes Neurais Convolucionais (CNN), tais como: VGG, ResNet, Inception e Xception.
- Redes Neurais Recorrentes (RNN), tais como: LSTM e GRU.

2.2.1 *Redes Neurais Convolucionais (CNN)*

O *Deep Learning* é uma ferramenta muito poderosa devido à sua capacidade de lidar com grandes quantidades de dados. O interesse em usar camadas ocultas superou as técnicas tradicionais, principalmente no reconhecimento de padrões. Uma das redes neurais profundas mais populares são as Redes Neurais Convolucionais (CNN) (Ioannidou et al., 2017).

Uma rede convolucional típica (ilustrada na Figura Fig. 1) consiste em uma combinação de três tipos principais de camadas: camadas convolucionais (Conv), camadas de agrupamento (*Pool* ou *pooling*) e camadas totalmente conectadas (*Fully-Connected*) (Ioannidou et al., 2017).

Vários filtros são usados para convoluir a imagem de entrada ou a saída da camada anterior reduzindo as imagens de entrada em um formato mais fácil de processar, sem perder recursos que são críticos para se obter uma boa previsão. Em seguida, os valores de saída desta operação passam por uma função de ativação de camada oculta (ex. ReLU, Sigmoid e Tanh) e posteriormente, alguma forma de

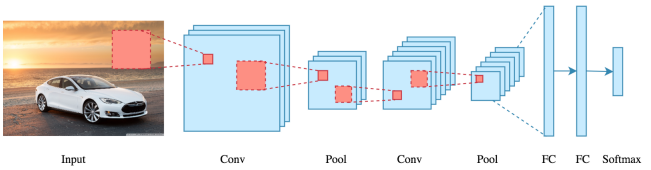

Figura 1: CNN (Fonte: Analytics Vidhya)

agrupamento é aplicada resultando em um número igual de mapas de características que são dados como entrada para a próxima camada (Ioannidou et al., 2017).

Na pilha de camadas convolucional e de *pooling*, uma ou mais camadas FC são adicionadas. Cada camada convolucional ou FC está relacionada a parâmetros/pesos específicos que devem ser aprendidos. As camadas de *pooling* são usadas para reduzir o tamanho espacial dos mapas de características e, portanto, preservar todas as informações importantes. As camadas FC realizam a multiplicação linear da entrada com a matriz de pesos e contêm o maior número de parâmetros em uma rede, portanto, treiná-los é computacionalmente caro. Por fim, a função *Softmax* transforma as saídas para cada classe em valores entre 0 e 1. Isso essencialmente dá a probabilidade de a entrada estar em uma determinada classe (Ioannidou et al., 2017).

2.2.2 *Redes Neurais Recorrentes (RNN)*

Uma RNN é uma arquitetura poderosa, normalmente usada para modelar dados sequenciais, como textos, sons e ações em vídeos. A sua maior diferença em relação as Redes Neurais Diretas (*Feed-forward Neural Network*) é o fato de possuírem memória interna que lembra sua entrada, tornando-as competentes para problemas envolvendo dados sequenciais em aprendizado de máquina. O seu grande diferencial é usar os mesmos pesos para cada elemento da sequência, diminuindo o número de parâmetros e permitindo que o modelo generalize para sequências de comprimentos variados (ver Figura Fig. 2). Em uma rede neural padrão, todas as entradas e saídas são independentes umas das outras, no entanto, em certos casos, como prever a próxima palavra da frase, as palavras anteriores são essenciais (Ioannidou et al., 2017).

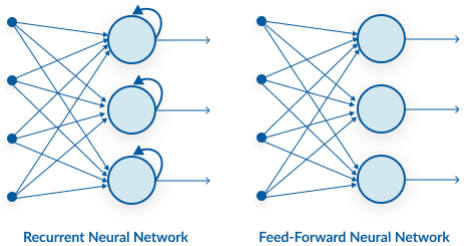

Figura 2: RNN vs. FFNN (Fonte: Analytics Vidhya)

Porém, um dos grandes problemas das RNN's eram problemas de explosão (gradiente com valores exorbitantes) e dissipação de gradiente (gradiente aproximadamente 0) durante a retro-propagação (*backpropagation*). O motivo da explosão e da dissipação do gradiente era dado por conta da captura de informações relevantes e irrelevantes. A resolução deste problema foi obtida por modelos que podem decidir e lembrar quais informações de uma entrada são relevantes, descartando todas as informações irrelevantes (Cuevas et al., 2020).

Isto é obtido utilizando-se portas (*Gates*). O LSTM (*Long-short-term memory*) e o GRU (*Gated Recurrent Unit*) possuem *Gates* como mecanismo interno, que controlam quais informações manter e quais informações descartar. As redes GRU são uma versão simplificada do LSTM, que podem alcançar resultados semelhantes ao LSTM, utilizando menos parâmetros e também são capazes de resolver os problemas de explosão e dissipação de gradiente (Cuevas et al., 2020).

2.3 Revisão Sistemática da Literatura

Para embasamento teórico e aplicado do estado da arte na área de pesquisa deste trabalho, foi realizada uma revisão sistemática da literatura (RSL) sobre o assunto. Foram realizadas quatro etapas da RSL, de acordo com o modelo proposto em Mariano et al. (2017). Tais etapas consistem em: a) definição de protocolo, b) coleta de referências, c) avaliação dos dados, e d) interpretação dos achados. As Seção 2.3.1 à Seção 2.3.4 detalham a realização destes passos.

2.3.1 *Etapa A: Definição de protocolo*

Segundo Nakagawa et al. (2017), na definição de protocolo deve ser preenchida uma tabela segmentada em cinco etapas (conforme Fig. 3):

1. Informações gerais;
2. Questão de pesquisa;
3. Identificando o estudo;
4. Seleção de avaliação do estudo;
5. Síntese dos dados.

1-Informações Gerais	
Título	Categorização de ações em vídeos de futebol utilizando uma arquitetura CNN-RNN
Autores	Matheus de Sousa Macedo e Diana Adamatti
Descrição	Com a expansão da tecnologia no futebol e pela melhora contínua do esporte, sendo uma área em crescimento e com enorme campo de busca para novas soluções
Objetivo	Classificador capaz de distinguir determinadas ações em um trecho de vídeo de uma partida de futebol

2-Questão de Pesquisa
Qual o estado da arte para classificadores de ações em vídeos de futebol utilizando Deep Learning?

3-Identificando o Estudo	
Palavra-Chave	Classifier, Categorization, RNN, CNN, GRU, Soccer Actions
Strings de Busca	"Classifier OR Categorization" AND "CNN" AND "RNN OR GRU" AND "Soccer Actions"
Critérios de Seleção das Bases	Bases relacionadas as áreas de computação
Bases Bibliográficas	IEEE, Springer, Scholar, Elsevier
Estratégia de Busca	Uso das strings de busca a partir dos periodicos de CAPES de acordo com os critérios de inclusão e exclusão

4-Seleção de Avaliação do Estudo	
Critério de Inclusão	• Ano (últimos 5 anos 2017-2022) • Língua (português ou inglês) • Ter a disponibilidade de download
Critério de Exclusão	• Não ter validação • Não informar o modelo/dataset utilizado • Evitar artigos de opinião ou sites
Estratégia de seleção	1º) Análise do título com as palavra-chave; 2º) Leitura de Título e abstract; 3º) Leitura Diagonal (título, abstract, introdução, legendas de figuras e tabelas, conclusão); 4º) Leitura completa.
Avaliação de qualidade	Quantidade de ações, Linguagem de programação, Dataset público, Tamanho do Dataset, Técnica de Deep Learning, Acurácia

5-Síntese dos Dados	
Extração dos Dados	Utilizando a string definida no portal da CAPES, exportar o BIBtex e importar no Mendley
Sumarização dos Dados	Utilizando os criterios de avaliação de qualidade, os artigos selecionados ao final da quarta etapa da estratégia de seleção (leitura completa) serão analisados

Figura 3: Protocolo utilizado na Revisão Sistemática

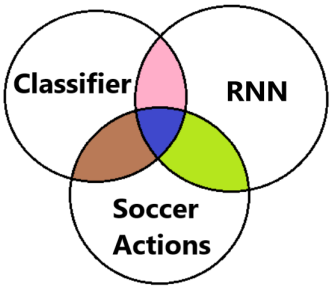

Figura 4: Diagrama de Venn das Áreas Correlacionadas a Pesquisa

Com as informações das tabelas estabelecidas, iniciou-se buscas genéricas chaveadas em quatro diferentes buscadores de trabalhos científicos consolidados: IEEEXplore, Google Scholar, Scopus (Elsevier) e Springer.

Também nesta etapa, foi definido o diagrama de Venn das áreas correlacionadas a pesquisa (Fig. 4). A ideia com a RSL é localizar os trabalhos que se enquadrem na zona "azul" da imagem.

O objetivo desta etapa foi encontrar na literatura classificadores de ações em vídeos que tenham sidos aplicados ao futebol, a fim de se obter respostas à questão principal do projeto: **"Qual o estado da arte para classificadores de ações em vídeos de futebol utilizando *Deep Learning*?"**. A Fig. 3 apresenta as palavras-chaves que foram utilizadas em cada uma das bases. O resultado de pesquisa de cada base foi exportado no formato BIBtex para posterior importação no software de gerenciamento de referências Mendeley.

Após as pesquisas, e com um grande número de artigos relacionados em mãos, foi-se utilizado os critérios inclusivos e exclusivos da Fig. 3 para que fosse possível prosseguir com a seleção. Foi decidido que seriam critérios de inclusão: o ano de publicação, ou seja, a relevância temporal do trabalho; a língua utilizada, limitadas ao inglês e ao português (idiomas que os autores compreendem); e a disponibilidade de download do trabalho. Da mesma forma, três critérios de exclusão foram definidos: se o artigo não possui validação; não informar o modelo e/ou dataset utilizado; e se é um artigo de opinião.

2.3.2 *Etapa B: Coleta de referências*

O principal objetivo desta etapa consiste em reunir um conjunto inicial de referências que possam ser interessantes, a partir dos resultados de busca obtidos na etapa A, filtrados pelos critérios de inclusão e exclusão definidos.

Desta maneira, aplicando os critérios sob os artigos de maior relevância na pós-busca e removendo os artigos duplicados, foram selecionados 80 artigos diferentes, sendo 6 oriundos da IEEEXplore, 26 do Google Scholar, 16 da Scopus(Elsevier) e, por fim, 32 da Springer (conforme Fig. 5).

2.3.3 *Etapa C: Avaliação dos Dados*

Dos 80 artigos encontrados, 40 foram selecionados para prosseguir para próxima etapa adiante, por meio da leitura dos títulos e das palavras-chave dos trabalhos.

A relevância dos títulos à proposta foi julgada com base

Bases	Quantidade
IEEE	6
Scholar	26
Scopus(Elsevier)	16
Springer	32
Duplicados	52
Total sem duplicados	80

Figura 5: Resumo dos trabalhos localizados por base

em 4 critérios não-exclusivos, sem quaisquer restrições de idioma, definidos pelos autores:

- Títulos com "*Video Classifier*";
- Títulos com "*RNN*";
- Títulos com "*Soccer Actions*";
- Títulos com maior semelhança à proposta.

Com 40 artigos restantes, iniciou-se a etapa de leitura dos resumos, ou *abstract*, para reduzir o tamanho desta amostra para um total de 20 artigos. Quanto mais compatível se mostrava a ideia geral do trabalho, maiores eram as chances de ele ser um dos selecionados para o próximo passo. Artigos que não envolviam "*Video Classifier*" ou "*Soccer Actions*" foram os primeiros a serem desconsiderados, enquanto os 20 artigos escolhidos se correlacionavam fortemente com a proposta, sendo até alguns deles tratando-se de Classificadores de ações em vídeos de futebol. Da mesma forma, artigos que se aprofundavam em técnicas específicas bastante distintas ou classificadores apenas de imagens também foram descartados.

Dos 20 artigos restantes, 11 foram escolhidos após a leitura diagonal. Por fim, 8 foram escolhidos para serem as referências teóricas de fato. Os critérios para exclusão ou inclusão dos artigos se resumiram ao quão específico cada artigo era na sua abordagem, e se tal abordagem era ou não suficientemente compatível com a proposta deste trabalho. Foram excluídos artigos que se utilizavam muito especificamente de técnicas distintas, bem como artigos muito generalistas que pouco viriam agregar em nível técnico.

Ao final da leitura completa e avaliação sob os critérios citados, foram selecionados por definitivo os artigos:(Baccouche et al., 2010),(Sen and Deb, 2022),(Cuevas et al., 2020),(Sanabria et al., 2022), (Deliege et al., 2021), (Ganesh et al., 2019),(Liu et al., 2017) e (Agyeman et al., 2019). De forma a compor a base teórica deste trabalho. Todos passaram por uma leitura completa por parte dos autores, de modo a compreender e interpretar os achados nos trabalhos, o que integra a etapa D da RSL.

A Fig. 6 apresenta um resumo desta etapa.

2.3.4 *Etapa D: Interpretação dos resultados*

Basicamente é a última etapa da RSL, podendo ser ou não realizada paralelamente com a leitura completa do artigo (que foi executada no passo anterior), e é dividida em 6 sub-etapas:

1. Agrupamento das referencias;
2. Sumarização dos tópicos principais;
3. Comparação entre os resultados (tabelas de características);
4. Coleta de dados;
5. Avaliação da sua RSL;

Fase	Quantidade
Total sem duplicados	80
1º) Análise do título com as palavras-chave	40
2º) Leitura de Título e abstract	20
3º) Leitura Diagonal (título, abstract, introdução, legendas de figuras e tabelas, conclusão)	11
4º) Leitura completa	8

Figura 6: Estratégia de seleção

6. Narração (escrita do artigo sobre a RSL ou de uma seção em um artigo).

Decidimos fazer paralelamente com o final do passo C, preenchendo uma ficha de leitura (Fig. 7) para facilitar a coleta de informações de cada artigo de referência para que a leitura não fique vaga e se esqueça, dessa forma ao lermos novamente os artigos, a compreensão acaba ficando mais fácil.

A tabela de características está contida na ficha de leitura, com duas características relevantes: Métodos (Modelo) e Resultados (Acurácia). A ficha compreende as 3 primeiras sub-etapas.

Na etapa 4, foram coletados os dados do Dataset SoccerNet-v2 (Deliege et al., 2021) utilizado como Dataset base deste trabalho. E as etapas 5 e 6 são a descrição que foi feita nesta seção.

2.3.5 *Informações relevantes à proposta deste trabalho*

O artigo Cuevas et al. (2020) contém o estado da arte de técnicas e aplicações que envolvem a classificação de ações em vídeos de futebol. É uma boa base para compreender as aplicações de Inteligência Artificial que podem ser feitas para o futebol. Sanabria et al. (2022) é um trabalho que tem como objetivo retirar as ações mais importantes de uma partida completa de futebol, com o intuito de gerar os melhores momentos das partidas. Ganesh et al. (2019) , Liu et al. (2017) , Agyeman et al. (2019) , Baccouche et al. (2010) e Sen and Deb (2022) são aplicações de classificadores de ações em vídeos de futebol, com diferentes técnicas e datasets. Deliege et al. (2021) é o Dataset base utilizado neste trabalho, como foi citado na seção anterior.

3 Metodologia Utilizada

Esta seção são apresentadas as etapas necessárias para a realização deste trabalho, focando na reestruturação do Dataset SoccerNet-v2 a fim de reduzir o Poder Computacional mínimo exigido, bem como na construção e na configuração da Rede Neural Híbrida CNN-RNN, capaz de categorizar ações distintas de futebol.

3.1 Dataset Utilizado

3.1.1 *SoccerNet-v2*

SoccerNet é um conjunto de dados em larga escala para compreensão de vídeos de futebol. Ele evoluiu ao longo dos anos para incluir várias tarefas, como detecção de ação, calibração da câmera, reidentificação e rastreamento de jogadores. É composto por 550 jogos de futebol completos transmitidos e 12 jogos de câmera única retirados das principais ligas européias (Premier League, La Liga, Bundesliga, Serie A, Ligue 1 e Champions League). Além disso, a equipe SoccerNet promove desafios anuais, onde as melhores equipes competem em nível internacional, a fim de expandir e melhorar o conhecimento na área de compreensão de vídeos de futebol.

Para se ter acesso ao *Dataset*, deve-se preencher um termo de "Acordo de não divulgação" (*NDA*), onde será recebido uma senha para fazer o *download* completo do *Dataset*. Todas as informações encontram-se no site oficial (Deliege et al., 2021) .

3.1.2 *Reestruturando o SoccerNet-v2*

O Dataset SoccerNet-v2 é estruturado da seguinte maneira:

- A camada de pastas superior corresponde a que liga aqueles vídeos foram retirados (Premier League, Ligue 1, Bundesliga, LaLiga, etc).
- A camada intermediária corresponde a temporada (14-15, 15-16, etc).
- E finalmente, a camada inferior mostra as informações da partida (data, hora e placar) e as partidas são divididas em dois vídeos (primeiro e segundo tempo) no formato *.mkv* acompanhado de um arquivo *.json* com os respectivos rótulos de ações (*labels*).

A Fig. 8 ilustra a explicação anterior:
A reestruturação foi divida em três etapas:

1. Cortes (através de um *script* Python) de tamanhos variáveis de cada partida completa dividindo-a por ações (Fig. 9).
2. Verificação manual se o corte realmente correspondia a ação anotada.
   O objetivo desta etapa era ter 500 vídeos confiáveis de cada uma das diferentes ações anotadas. Desta forma, decidiu-se escolher 4 ações para serem verificadas manualmente, para obter-se 500 vídeos confiáveis, onde se escolheu as seguintes ações:
   1. *Clearance* (Tiro de meta).
   2. *Direct free-kick* (Tiro livre direto).
   3. *Kick-off* (Início/Re-início de jogo).
   4. *Yellow card* (Cartão amarelo).

   A Fig. 10 ilustra esta etapa:
3. Desta maneira, pude criar o meu próprio Dataset anotado com a confiabilidade necessária. 2000 vídeos no total sendo 1800 para treino e 200 para teste (Fig. 11).

Algumas limitações impossibilitaram a utilização de mais ações:

- **Poder computacional**: O autor utilizava o Google Colab (serviço de armazenamento em nuvem de notebooks voltados à criação e execução de códigos em Python, diretamente de um navegador). E por ser um serviço gratuito, ele limita o usuário a utilizar o serviço por uma quantidade de horas por dia.
- **Inviabilidade**: assistir manualmente mais de 500 ví-

Texto 1	
Título	Techniques and applications for soccer video analysis: A survey (2020)
Autores	Carlos Cuevas, Daniel Quilon and Narciso Garcia (Espanha)
Objetivo	Levantamento de técnicas e aplicações mais significativas que foram propostas ao longo das últimas duas décadas para analisar sequências de vídeos de futebol
Métodos	Estratégias de detecção de eventos, Estratégias de detecção e rastreamento de jogadores e/ou bola e Estratégias de análise do jogo
Resultados	--
Relação	Estado da arte em classificar ações em vídeos de futebol

Texto 2	
Título	A Multi-stage deep architecture for summary generation of soccer videos (2022)
Autores	Melissa Sanabria, Frederic Precioso, Pierre-Alexandre Mattei, and Thomas Menguy (França)
Objetivo	Gerar automaticamente resumos em vídeo de partidas de futebol.
Métodos	LSTM MIL (Multiple Instance Learning) Pooling method
Resultados	--
Relação	Sumarizar ações mais importantes de futebol em videos

Texto 3	
Título	SoccerNet-v2: A Dataset and Benchmarks for Holistic Understanding of Broadcast Soccer Videos (2021)
Autores	Adrien Deliege, Anthony Cioppa, Silvio Giancola, Meisam J. Seikavandi, Jacob V. Dueholm, Kamal Nasrollahis, Bernard Ghanem, Thomas B. Moeslund, Marc V. D. (Arábia Saudita)
Objetivo	Dataset anotado de partidas de futebol.
Métodos	Vídeos de cada tempo (+-45 min) de partidas das principais ligas (Premier League, La Liga, Bundesliga, Serie A, Champions League e Ligue 1)
Resultados	17 classes
Relação	Dataset que pode ser utilizado para classificar ações

Texto 4	
Título	A novel framework for fine grained action recognition in soccer (2019)
Autores	Yaparla Ganesh, Allaparthi Sri Teja, Sai Krishna Munnangi & Garimella Rama Murthy (India)
Objetivo	Propor um dataset personalizado e modelos para classificar ações em videos de futebol.
Métodos	CNN3D, Inception + LSTM, GAWAC with σ2 = 0.5
Resultados	Dataset Soccer-8k com 6 classes e modelo com 62% de acurácia
Relação	Dataset e modelo que podem ser utilizados para classificar ações

Texto 5	
Título	Soccer video event detection using 3d convolutional networks and shot boundary detection via deep feature distance (2017)
Autores	Tingxi Liu, Yao Lu, Xiaoyu Lei, Lijing Zhang, Haoyu Wang,Wei Huang, and Zijian Wang (China)
Objetivo	Modelos para classificar ações em videos de futebol.
Métodos	CNN3D
Resultados	Dataset Soccer-152A com 8 classes e modelo com 81% de acurácia
Relação	Modelos que podem ser utilizados para classificar ações

Texto 6	
Título	Soccer video summarization using deep learning (2019)
Autores	Rockson Agyeman, Rafiq Muhammad and Gyu Sang Choi (Coréia do Sul)
Objetivo	Modelo capaz de classificar e anotar 5 ações em videos de futebol
Métodos	CNN3D + RNN(LSTM)
Resultados	Acurácia de 96%
Relação	Modelo que podem ser utilizados para classificar ações

Texto 7	
Título	Action Classification in Soccer Videos with Long Short-Term Memory Recurrent Neural Networks
Autores	Moez Baccouche, Franck Mamalet, Christian Wolf, Christophe Garcia, and Atilla Baskurt (França)
Objetivo	Classificar ações em vídeos de futebol utilizando uma arquitetura RNN
Métodos	BoW + dominant motion + RNN(LSTM)
Resultados	Acurácia de 92%
Relação	Classificar ações de futebol em videos

Texto 8	
Título	Categorization of actions in soccer videos using a combination of transfer learning and Gated Recurrent Unit (2022)
Autores	Anik Sen and Kaushik Deb (India)
Objetivo	Classificar diferentes comprimentos de videos de ações de futebol
Métodos	CNN(VGG16) + RNN(GRU)
Resultados	Acurácia de 94%
Relação	Classificar ações de futebol em videos

Figura 7: Ficha de leitura

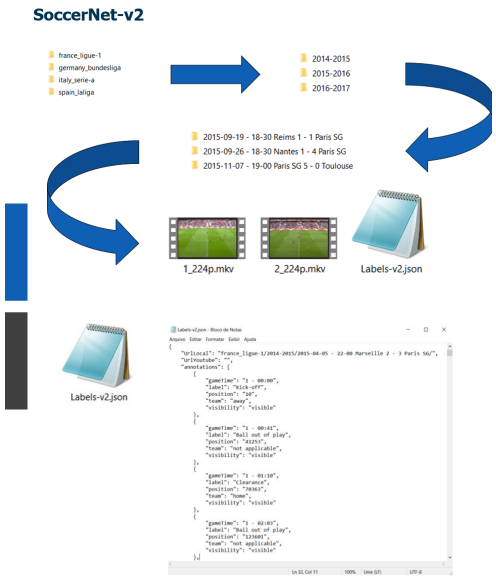

Figura 8: Estrutura inicial SoccerNet-v2

Figura 9: Cortes por ações

Figura 10: Reformulando o SoccerNet-v2

Figura 11: Criando um novo Dataset

deos para cada ação.

Um outro Dataset de teste foi criado da mesma forma do Dataset principal para verificar o comportamento da rede, e foi utilizado no Experimento 2. Ele tem a seguinte estrutura:

- 3 ações: *Corner* (Escanteio), *Penalty* (Cobranças de Penâlti) e *Substitution* (Substituição).
- 150 vídeos para cada ação.
- 450 vídeos no total, sendo 405 vídeos para treino e 45 para teste.

3.2 Rede Neural Utilizada

Diversos testes foram realizados manualmente até se chegar ao melhor modelo, foram utilizadas as CNN's VGG16, VGG19, InceptionV3 e Xception. E as RNN's GRU e LSTM. As arquiteturas utilizadas podem ser encontradas detalhadamente na API Python Keras.

O melhor resultado de teste tem a seguinte configuração: CNN InceptionV3 para extração de *features* mais importantes, juntamente com a RNN Gated Recurrent Unit (GRU). Os parâmetros utilizados são na Fig. 12).

```
IMG_SIZE = 320
BATCH_SIZE = 32
EPOCHS = 100
MAX_SEQ_LENGTH = 100
NUM_FEATURES = 2048
```

Figura 12: Hiperparâmetros

IMG_SIZE é a altura e largura da imagem que será extraída do vídeo.

BATCH_SIZE é a quantidade de amostras de treino utilizadas em uma época.

MAX_SEQ_LENGTH é a quantidade de quadros extraídos de cada vídeo.

NUM_FEATURES é o número de *features* que o modelo pré-treinado gera para cada quadro de entrada.

```
feature_extractor = keras.applications.InceptionV3(
    weights="imagenet",
    include_top=False,
    pooling="avg",
    input_shape=(IMG_SIZE, IMG_SIZE, 3),
)
```

Figura 13: Modelo CNN utilizado para a extração de *features*

O modelo para extração de *features* pré-treinado utilizado é o InceptionV3 com pesos do Dataset Imagenet-1k, a camada superior que classifica a imagem foi removida e também foi alterado o **input_shape** para **IMG_SIZE** (Fig. 13).

O modelo RNN é constituído da seguinte forma (Fig. 14):

- Camada (*layer*) GRU com 32 unidades e função de ativa-

```
x = keras.layers.GRU(32, return_sequences=True)(
    frame_features_input, mask=mask_input
)
x = keras.layers.GRU(20)(x)

x = keras.layers.Dense(2048, activation="relu")(x)
x = keras.layers.Dense(1024, activation="relu")(x)

x = keras.layers.Dense(512, activation="relu")(x)
x = keras.layers.Dense(512, activation="relu")(x)

x = keras.layers.Dense(256, activation="relu")(x)
x = keras.layers.Dense(256, activation="relu")(x)

x = keras.layers.Dense(128, activation="relu")(x)
x = keras.layers.Dense(128, activation="relu")(x)

output = keras.layers.Dense(len(class_vocab), activation="softmax")(x)

rnn_model = keras.Model([frame_features_input, mask_input], output)

rnn_model.compile(
    loss="sparse_categorical_crossentropy", optimizer="adam",
    metrics=["accuracy"]
)
```

Figura 14: Modelo RNN (GRU)

ção padrão 'tahn'.
- Camada (*layer*) GRU com 20 unidades e função de ativação padrão 'tahn'.
- Camada (*layer*) Densa com 2048 unidades e função de ativação 'relu'.
- Camada (*layer*) Densa com 1024 unidades e função de ativação 'relu'.
- Duas Camadas (*layers*) Densas com 512 unidades e função de ativação 'relu'.
- Duas Camadas (*layers*) Densas com 256 unidades e função de ativação 'relu'.
- Duas Camadas (*layers*) Densas com 128 unidades e função de ativação 'relu'.
- Camada (*layer*) Densa de saída com 4 unidades e função de ativação 'softmax', pois usaremos as probabilidades para uma classificação multiclasse.
- A função de perda '*sparse categorical crossentropy*', pois é um caso de classificação multiclasse, otimizador 'adam' e métrica selecionada para precisão.

3.3 Métricas de Avaliação

- **Acurácia**: indica a performance geral do modelo. Dentre todas as classificações, quantas o modelo classificou corretamente (Predições corretas / Total de predições).
- **Matriz de Confusão**: exibe a distribuição dos resultados em termos de suas classes verdadeiras e de suas classes previstas. Os respectivos rótulos (*labels*) são: 0-*Clearance*, 1-*Direct free-kick*, 2-*Kick-off*, 3-*Yellow card*.
- **F1-Score**: pode ser interpretado como uma média harmônica de precisão (*precision*) e sensibilidade (*recall*), sendo a precisão (*precision*) a proporção de identificações positivas que foram classificadas corretamente, e a sensibilidade (*recall*) é a proporção de positivos que foram identificados corretamente, ou seja, o quão bom o modelo é para classificar aquela determinada classe, onde um escore F1 atinge seu melhor valor em 1 e pior escore em 0. A contribuição relativa da precisão e sensibilidade para o escore F1 são iguais. A fórmula é dada por:

  F1 = 2 * (precision * recall) / (precision + recall)

- **Curva ROC**: é uma representação gráfica que ilustra o desempenho (ou performance) de um sistema classificador binário à medida que o seu limiar de discriminação varia. A curva ROC é também conhecida como curva de característica de operação relativa, porque o seu critério de mudança é resultado da operação de duas características (TPV e TPF).
  A curva ROC é obtida pela representação da taxa TPV = Positivos Verdadeiros / Positivos Totais versus a taxa TPF = Positivos Falsos / Negativos Totais, para vários valores do limiar de classificação. O TPV é também conhecido como sensibilidade (ou taxa de verdadeiros positivos), e TPF = 1-especificidade (ou taxa de falsos positivos). A especificidade é conhecida como taxa de verdadeiros negativos (TVN).
  O valor da área (AUC) varia de 0,0 até 1,0 e o limiar entre a classe é 0,5. Ou seja, acima desse limite, o algoritmo classifica em uma classe e abaixo em outra classe. Um modelo cujas previsões estão 100% erradas tem uma área (AUC) igual à 0, enquanto um modelo cujas previsões são 100% corretas tem uma área (AUC) igual à 1.

4 Resultados

Três grandes Experimentos foram feitos para analisar o comportamento da rede com diferentes ações, quantidade de vídeos e quantidade de ações.

No Experimento 1 utiliza-se uma grande quantidade de vídeos para cada ação (500 vídeos para cada) e 4 ações no total. Na Seção 4.4, foram feitas inferências para verificar se a rede possuía comportamentos tendenciosos a uma determinada ação.

No Experimento 2 utiliza-se uma quantidade menor de ações, 3, e uma quantidade menor de vídeos para cada ação (150 vídeos para cada).

E por fim, no Experimento 3 utiliza-se uma maior quantidade de ações, totalizando 7 ações, e também mantendo a quantidade de 150 vídeos para cada ação.

A Fig. 15 sumariza os Experimentos realizados neste Capítulo.

Experimento	Ações	Vídeos (Total)	Inferências?
Exp. 1	4	2000	4
Exp. 2	3	450	--
Exp. 3	7	1050	--

Figura 15: Sumarização dos Experimentos

4.1 Experimento 1 - Modelo principal

A partir do modelo rede neural escolhido e com as configurações anteriormente apresentadas, chegou-se ao melhor resultado de acurácia de 94,0%.

Para verificar o desempenho do modelo, foi construída a Matriz de Confusão, e além do cálculo de acurácia padrão (Predições corretas / Total de predições) , foram utilizadas outras métricas, como F1-Score e Curva ROC.

4.1.1 *Matriz de Confusão*

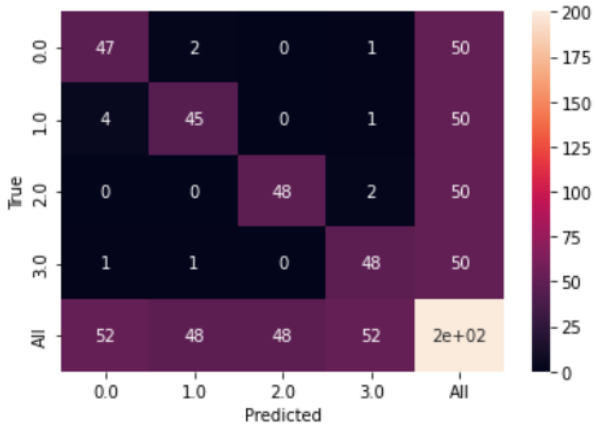

Figura 16: Experimento 1 - Matriz de confusão

Analisando a Matriz de Confusão (Fig. 16), pode-se perceber primeiramente que dos 200 vídeos totais, 188 foram classificados corretamente, o que corresponde a acurácia de 94%. Como dito anteriormente, foram testados 50 vídeos de cada classe.

Na classe *Clearance* (Tiro de meta), 47 vídeos foram classificados corretamente (Verdadeiros Positivos), 3 vídeos classificados incorretamente (Falsos Negativos) sendo 2 como *Direct free-kick* (Tiro livre direto) e 1 como *Yellow card* (Cartão amarelo).

Na classe *Direct free-kick* (Tiro livre direto) foi a classe que houve a menor taxa de acertos, o modelo previu 45 vídeos corretamente (Verdadeiros Positivos), e 5 vídeos incorretamente (Falsos Negativos) sendo 4 classificados como *Clearance* (Tiro de meta) e 1 como *Yellow card* (Cartão amarelo) .

Na classe *Kick-off* (Início/Re-início de jogo) teve 48 vídeos classificados corretamente (Verdadeiros Positivos) e 2 incorretamente (Falsos Negativos) classificados como *Yellow card* (Cartão amarelo).

A última classe *Yellow card* (Cartão amarelo) também teve 48 vídeos classificados corretamente (Verdadeiros Positivos) e 2 incorretamente (Falsos Negativos) sendo 1 como *Clearance* (Tiro de meta) e 1 como *Direct free-kick* (Tiro livre direto).

4.1.2 *F1-Score*

A métrica F1-Score mostra o balanço entre a *precision* e *recall* do modelo para cada uma das classes. Uma vez que seu valor está alto significa que a acurácia que se obteve é relevante, ou seja, os valores Verdadeiros Positivos, Verdadeiros Negativos, Falsos Positivos e Falsos Negativos aferidos não apresentam grandes distorções. Também pode-se interpretar como uma medida de confiabilidade da acurácia.

A Fig. 17 apresenta o F1-Score para cada classe do modelo, sendo todos os valores acima de 90%:

```
                Clearance  Direct free-kick    Kick-off  Yellow card
F1-Score: [0.92156863 0.91836735 0.97959184 0.94117647]
```

Figura 17: Experimento 1 - Métrica F1-Score

Dessa forma, conclui-se que o modelo tem uma acurácia relevante.

4.1.3 *Curva ROC*

Como explicado anteriormente, a Curva ROC tem o objetivo de analisar o poder preditivo de um modelo, garantindo que ele irá detectar o máximo possível de Verdadeiros Positivos, enquanto minimiza os Falsos Positivos. As Fig. 18 à Fig. 21 mostram os valores da Curva ROC para cada uma das classes.

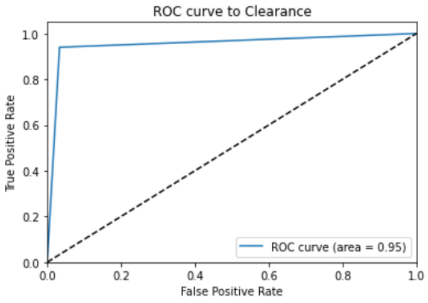

Figura 18: Experimento 1 - Curva ROC para *Clearance*

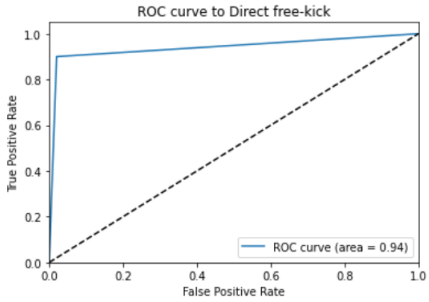

Figura 19: Experimento 1 - Curva ROC para *Direct free-kick*

O maior valor preditor é da classe *Kick-off* como esperado, com AUC = 0.98, e o menor valor preditor é da classe *Direct-free-kick* com AUC = 0.94, também já esperado. Dessa maneira, como todas as classes tem AUC's acima de 0.90, conclui-se que o modelo consegue distinguir satisfatoriamente cada uma das classes treinadas.

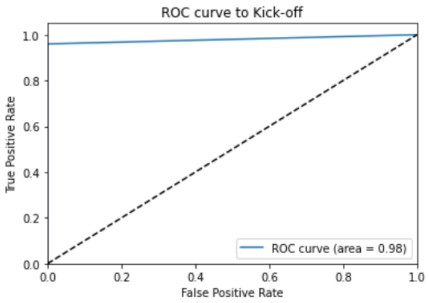

Figura 20: Experimento 1 - Curva ROC para *Kick-off*

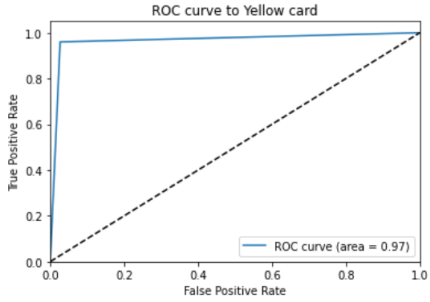

Figura 21: Experimento 1 - Curva ROC para *Yellow card*

4.2 Experimento 2 - Utilizando outras ações

Utilizando a mesma rede neural, com as mesmas configurações, e com o segundo Dataset especificado anteriormente:

- 3 ações: *Corner* (Escanteio), *Penalty* (Cobranças de Penâlti) e *Substitution* (Substituição).
- 150 vídeos para cada ação.
- 450 vídeos no total, sendo 405 vídeos para treino e 45 para teste.

Chegou-se ao resultado de acurácia de 88,89%.

4.2.1 *Matriz de Confusão*

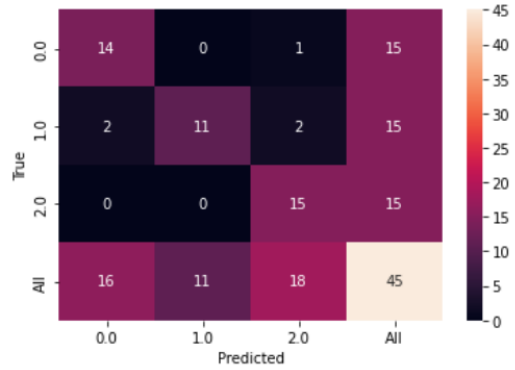

Figura 22: Experimento 2 - Matriz de confusão

Na Fig. 22, 0 corresponde a *Corner*, 1 a *Penalty* e 2 a *Substitution*.

A classe *Penalty* (Cobranças de Pênaltis) foi a classe que houve a menor taxa de acertos, o modelo previu 11 vídeos corretamente (Verdadeiros Positivos), e 4 vídeos incorretamente (Falsos Negativos) sendo 2 classificados como *Corner* (Escanteio) e 2 como *Substitution* (Substituição). Esse resultado ocorre pela limitação da quantidade de vídeos de Penâlti retirados do Dataset. Por esse motivo, não se pode ter o mesmo rigor de seleção feito nas outras ações que tinham quantidades muito maiores de vídeos.

4.2.2 *F1-Score*

```
                Corner          Penalty         Substitution
F1-Score: [0.90322581 0.84615385 0.90909091]
```

Figura 23: Experimento 2 - F1-Score

O modelo mantêm resultados satisfatórios para a métrica F1, mesmo com poucos vídeos, tanto *Corner* quanto *Substitution* obtiveram valores acima de 90% de acurácia (Fig. 23).

4.2.3 *Curva ROC*

As Figs. 24 a 26 apresentam os valores da curva ROC para este experimento. O maior valor preditor é da classe *Substitution* como esperado, com AUC = 0.95, e o menor valor preditor é da classe *Penalty* com AUC = 0.87, também já esperado. Dessa maneira, conclui-se que o modelo consegue distinguir satisfatoriamente cada uma das classes treinadas.

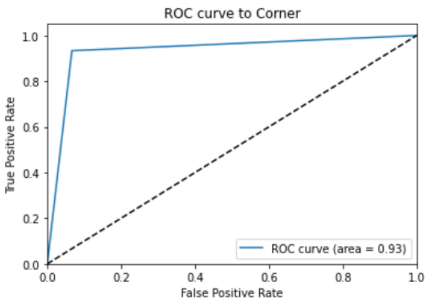

Figura 24: Experimento 2 - Curva ROC para *Corner*

4.3 Experimento 3 - Utilizando sete ações

Utilizando a mesma rede neural, com as mesmas configurações, construímos um terceiro Dataset com todas as sete ações anteriores:

- 7 ações: *Clearance* (Tiro de meta), *Corner* (Escanteio), *Direct free-kick* (Tiro livre direto), *Kick-off* (Início/Reinício de jogo), *Penalty* (Cobranças de Penâlti), *Substi-*

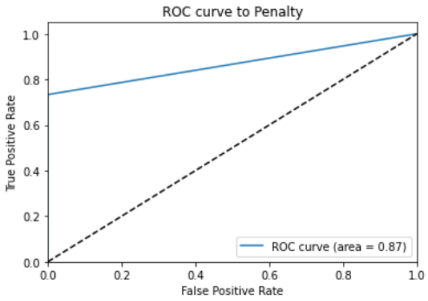

Figura 25: Experimento 2 - Curva ROC para *Penalty*

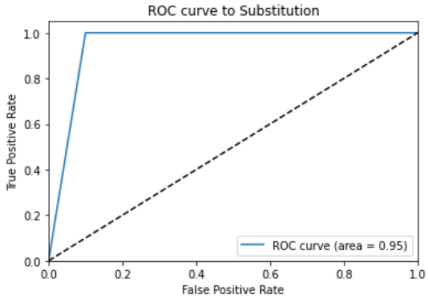

Figura 26: Experimento 2 - Curva ROC para *Substitution*

tution (Substituição) e *Yellow card* (Cartão amarelo).

- 150 vídeos para cada ação.
- 1050 vídeos no total, sendo 945 vídeos para treino e 105 para teste.

Chegou-se ao resultado de acurácia de 86,67%.

4.3.1 *Matriz de Confusão*

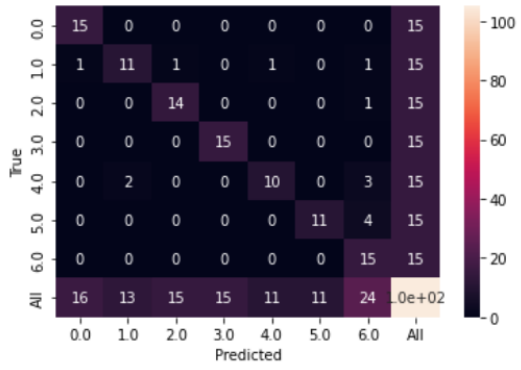

Figura 27: Experimento 3 - Matriz de confusão

Na Fig. 27, 0 corresponde a *Clearance*, 1 a *Corner*, 2 a *Direct-free-kick*, 3 a *Kick-off*, 4 a *Penalty*, 5 a *Substitution* e 6 a *Yellow-card*.

A classe *Penalty* (Pênalti) foi a classe que houve a menor taxa de acertos, o modelo previu 10 vídeos corretamente (Verdadeiros Positivos), e 5 vídeos incorretamente (Falsos Negativos). As classes *Clearance* (Tiro de meta), *Kick-off* (Início/Re-início de jogo) e *Yellow card* (Cartão amarelo) foram as classes no qual a rede acertou todos os vídeos.

4.3.2 *F1-Score*

```
                Clearance     Corner  Direct-free-kick  Kick-off    Penalty  Substitution
F1-Score: [0.96774194 0.78571429 0.93333333 1.0 0.76923077 0.84615385
 Yellow-card
0.76923077]
```

Figura 28: Experimento 3 - F1-Score

O modelo mantêm resultados satisfatórios para a métrica F1 exceto para a classe *Yellow-card* (Cartão amarelo), no qual pode-se notar uma distorção, ou seja, a métrica mostra que o modelo acaba tendendo a esta classe quando está em dúvida. O ângulo da câmera dos vídeos é o principal fator para essa anomalia, visto que os vídeos de Cartão amarelo são vídeos com um nível de zoom muito alto, e quando um vídeo de outra classe têm essa característica, a rede têm dificuldade em determinar a classe correta (Fig. 28).

4.3.3 *Curva ROC*

As Figs. 29 a 35 apresentam os valores da curva ROC para este experimento.O maior valor preditor é da classe *Kick-off* como esperado, com AUC = 1.0, e o menor valor preditor é da classe *Penalty* com AUC = 0.83, também já esperado. Dessa maneira, conclui-se que o modelo consegue distinguir as classes treinadas, pórem para as classes *Penalty*, *Substitution* e *Yellow-card*, que possuem características muito semelhantes, o modelo deve ser alimentado com mais dados para conseguir distinguir essas classes satisfatoriamente.

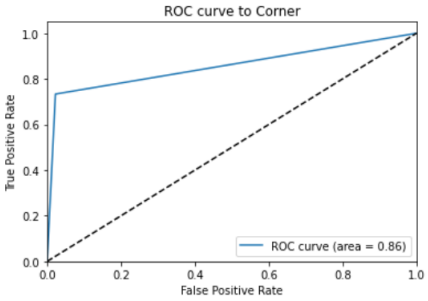

Figura 29: Experimento 3 - Curva ROC para *Corner*

4.4 Inferências

De forma a verificar o comportamento da rede principal (rede treinada com 4 ações e 500 vídeos de cada ação), foram feitas inferências. O objetivo das inferências foi analisar como o modelo encontrado pela rede principal se comportaria para ações similares (Ação de futebol não

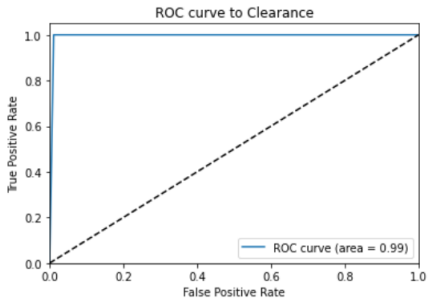

Figura 30: Experimento 3 - Curva ROC para *Clearance*

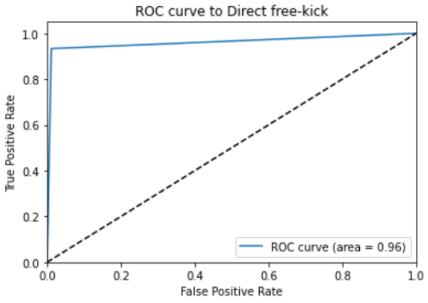

Figura 31: Experimento 3 - Curva ROC para *Direct-free-kick*

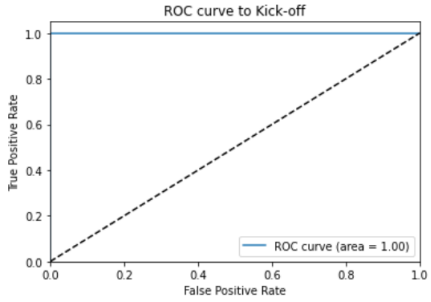

Figura 32: Experimento 3 - Curva ROC para *Kick-off*

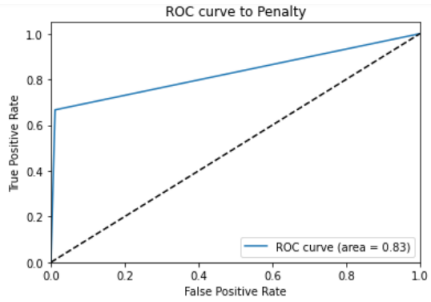

Figura 33: Experimento 3 - Curva ROC para *Penalty*

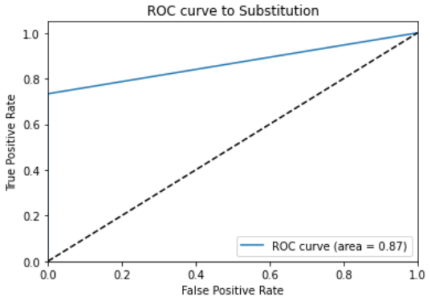

Figura 34: Experimento 3 - Curva ROC para *Substitution*

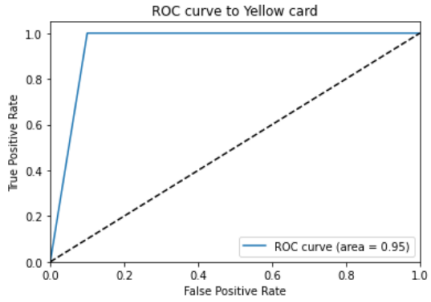

Figura 35: Experimento 3 - Curva ROC para *Yellow-card*

vista, Ação de esporte semelhante) ou não (Ações de outros esportes, como golfe e skate). Desta forma, validando as ações que o modelo aprendeu ou analisando as características semelhantes das ações.

4.4.1 *Vídeo de uma ação de futebol não vista*

Utilizando um vídeo correspondente a um Pênalti, a rede inferiu 99.67% de certeza que era a ação *Direct free-kick* (Tiro livre direto). Analisando esse resultado, percebe-se que a rede soube diferenciar entre as diferentes ações, e escolheu a ação que realmente mais se assemelha a um Pênalti (Fig. 36).

4.4.2 *Vídeo de uma ação de um esporte semelhante (futsal)*

Utilizando um vídeo correspondente a um Pênalti no futsal, a rede inferiu 57.08% de certeza que era a ação *Yellow card* (Cartão amarelo), porém identificou semelhança também com a ação *Direct free-kick* inferindo 23.36%. A falta da cor verde do gramado e o ângulo da câmera aproximado, fizeram com que a rede ficasse em dúvida e considerando essas condições entendemos o motivo da escolha da rede (Fig. 37).

4.4.3 *Vídeos de outros esportes (golfe e skate)*

Utilizando um vídeo de uma tacada de golfe, a rede inferiu 82.38% de certeza que o vídeo correspondia a ação *Clearance* (Tiro de meta), analisando esse resultado percebe-se que a rede soube diferenciar entre as diferentes ações, e escolheu a ação que realmente mais se assemelha a uma tacada de Golfe (Fig. 38).

Utilizando um vídeo de Skate, a rede inferiu 83.10% de certeza que o vídeo correspondia a ação *Clearance* (Tiro

Figura 36: Inferência: ação não vista

Figura 37: Inferência: esporte semelhante

Figura 38: Inferência: outros esportes (golfe)

Figura 39: Inferência: outros esportes (skate)

de meta), o ângulo da câmera aproximado (*zoom*) trouxe essa conclusão para a rede (Fig. 39).

5 Conclusões

Os objetivos deste trabalho era propor uma reformulação do Dataset SoccerNet e um classificador CNN-RNN com acurácia maior que 90% para classificar ações em vídeos de futebol. Esta foi uma tarefa árdua, que teve diversas dificuldades durante a sua construção. Mesmo o futebol sendo um dos esportes mais populares, lucrativos e assistidos do mundo, ainda há poucos trabalhos com esse embasamento. A maior aplicabilidade deste embasamento é na geração de destaques, cálculos estatísticos e resumo das partidas de futebol, utilizados por emissoras de transmissão e também por empresas de apostas esportivas.

Durante essa construção, surgiu o desafio de reformular o Dataset SoccerNet, para atender a limitação computacional do trabalho. Essa escolha trouxe uma outra limitação, a de se assistir vídeo a vídeo a fim de se obter um *dataset* totalmente revisado e mais confiável. Nesta tarefa, foi construído um *dataset* com 4 ações e 500 vídeos de cada ação totalmente revisado. Também foi construído um segundo *dataset* com mais 3 ações e 150 vídeos de cada ação, também totalmente revisado.

Em relação a rede, ao categorizar-se ações de futebol deve-se considerar diversas condições de captura (por exemplo, mudança brusca no ângulo da câmera), que muitas vezes impossibilita uma distinção conclusiva entre as diferentes ações do futebol, além de que muitas ações são muito semelhantes. Utilizou-se diversas configurações de CNN's (VGG16, VGG19, InceptionV3 e Xception) e RNN's (GRU e LSTM) até chegar na melhor configuração, que foi a CNN InceptionV3 e a RNN GRU. Mesmo com essas dificuldades, a rede proposta obteve 94% de acurácia em um Dataset Anotado e totalmente revisado, composto por 4 ações, conforme a Seção 4.1.

Para trabalhos futuros, deixamos algumas sugestões:

- Recomenda-se a utilização de mais poder computaci-

onal, que foi uma das limitações deste trabalho, com isso é possível a ampliação do Dataset, com mais ações. Tem-se como ideia que a quantidade de 500 vídeos para cada ação seja a quantidade ideal.
- Outro passo que pode ser desenvolvido é a utilização de uma ferramenta de auto *tuning* de hiperparâmetros, para buscar encontrar configurações ainda melhores dos hiperparâmetros da rede.
- Pode-se tentar utilizar outras arquiteturas de CNN's, tais como: ResNet's e EfficientNet's.
- Por fim, a partir desse modelo, pode-se montar uma arquitetura para geração de destaques e melhores momentos.

Referências

Agyeman, R., Muhammad, R. and Choi, G. S. (2019). Soccer video summarization using deep learning, pp. 270–273. https://doi.org/10.1109/MIPR.2019.00055.

Baccouche, M., Mamalet, F., Wolf, C., Garcia, C. and Baskurt, A. (2010). Action classification in soccer videos with long short-term memory recurrent neural networks, *in* K. Diamantaras, W. Duch and L. S. Iliadis (eds), *Artificial Neural Networks – ICANN 2010*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 154–159. https://doi.org/10.1007/978-3-642-15822-3_20.

Cuevas, C., Quilón, D. and Garcia, N. (2020). Techniques and applications for soccer video analysis: A survey, *Multimed. Tools Appl.* **79**: 29685–29721. https://doi.org/10.1007/s11042-020-09409-0.

Deliege, A., Cioppa, A., Giancola, S., Seikavandi, M. J., Dueholm, J. V., Nasrollahis, K., Ghanem, B., Moeslund, T. B. and D., M. V. (2021). Soccernet-v2: A dataset and benchmarks for holistic understanding of broadcast soccer videos. https://doi.org/10.1109/CVPRW53098.2021.00508.

Ganesh, Y., Teja, A. S., Munnangi, S. K. and Murthy, G. R. (2019). A novel framework for fine grained action recognition in soccer, pp. 137–150. https://doi.org/10.1007/978-3-030-20518-8_12.

Ioannidou, A., Chatzilari, E., Nikolopoulos, S. and Kompatsiaris, I. (2017). Deep learning advances in computer vision with 3d data: A survey, *ACM Computing Surveys* **50**. http://dx.doi.org/10.1145/3042064.

Liu, T., Lu, Y., Lei, X., Zhang, L., Wang, H., Huang, W. and Wang, Z. (2017). Soccer video event detection using 3d convolutional networks and shot boundary detection via deep feature distance, pp. 440–449. https://doi.org/10.1007/978-3-319-70096-0_46.

Mariano, D., Leite, C., Santos, L., Rocha, R. and Melo-Minardi, R. (2017). A guide to performing systematic literature reviews in bioinformatics, *arXiv: Quantitative Methods* . Dhttps://doi.org/10.48550/arXiv.1707.05813.

Nakagawa, E., Scannavino, K., Fabbri, S. and Ferrari, F. (2017). *Revisão Sistemática da Literatura em Engenharia de Software: Teoria e Prática*, Elsevier Brasil. Available at https://books.google.com.br/books?id=kCspDwAAQBAJ.

Sanabria, M., Precioso, F., Mattei, P.-A. and Menguy, T. (2022). A multi-stage deep architecture for summary generation of soccer videos, *arXiv preprint arXiv:2205.00694* . https://doi.org/10.48550/arXiv.2205.00694.

Sen, A. and Deb, K. (2022). Categorization of actions in soccer videos using a combination of transfer learning and gated recurrent unit, *ICT Express* **8**(1): 65–71. https://doi.org/10.1016/j.icte.2021.03.004.

Szeliski, R. (2010). *Computer Vision: Algorithms and Applications*, 1st edn, Springer-Verlag, Berlin, Heidelberg. https://doi.org/10.1007/978-3-030-34372-9.