Utilizando técnicas de visão computacional para reconhecimento de ações em vídeos de futebol

Gabriel Rocha Martins
Universidade Federal de Minas Gerais
Belo Horizonte, Brasil
garoma20@ufmg.br

Erickson Rangel do Nascimento
Universidade Federal de Minas Gerais
Belo Horizonte, Brasil
erickson@dcc.ufmg.br
Orientador

Resumo

Este trabalho propõe SoccerEventSpotNet, um arcabouço para produção de soluções para a tarefa de classificação automática de tipos de eventos decorridos em vídeos de futebol. Além do repositório gerado, o projeto apresenta duas outras contribuições principais: a criação de um banco de dados de vídeos de jogos de futebol associados à dados de evento com 55 vídeos de jogos completos que referenciam mais de 90 mil eventos únicos e a definição da tarefa de classificação de vídeos de futebol dentre tipos de eventos pré-definidos tanto dentre 10 classes, onde foi gerado um modelo com acurácia de 0.693 e acurácia no top 3 de 0.946, quanto dentre 36 classes, onde for gerado um modelo com acurácia de 0.517 e acurácia no top 3 de 0.777, determinando um baseline para a tarefa definida.

1. Introdução

O futebol é um esporte que surgiu na Inglaterra durante o século 19 e que hoje é jogado no mundo todo. Segundo relatório anual da FIFA (Federação Internacional de Futebol) [2] o esporte gerou mais de 480 milhões de doláres apenas no ano de 2024, sendo grande parte desta renda advinda dos direitos de transmissão dos campeonatos organizados e direitos sobre campanhas publicitárias. Sendo assim, apesar de ser a princípio uma ativadade de entretenimento o setor se tornou relevante no cenário internacional dada a grande movimentaçao financeira gerada pelos jogos, o que fez com que a enfoque no mercado aumentasse à medida que empresas de diversos setores viram no entretenimento um meio de alavancar seus negócios com campanhas publicitárias e afins.

Dado o crescimento do esporte como um todo, a competitividade aumentou à medida que bons resultados poderiam resultar em grandes contratos tanto para os jogadores, quanto para os clubes. Neste contexto, a busca por maneiras eficientes de melhorar o desempenhos das equipes tornou o uso das imagens de transmissões de jogos fontes de dados que poderiam ser utilizados para geração de estatísticas, análises táticas, análises técnicas, entre outras tarefas que pudessem cooperar com o aumento da performance das equipes. Com isso, houve o surgimento de empresas como Wyscout e Statsbomb que tornaram a produção de dados esportivos o seu negócio, tendo como produto principal a produção de dados que auxiliam em análises baseadas em dados esportivos. Dentre os dados que podem ser gerados para cada jogo, os dados de eventos são fontes relevantes para a produção de diversos tipos de análises tanto do comportamento conjunto das equipes, quanto do individual de cada jogador, uma vez que ele apresenta, para cada evento que ocorre durante uma partida, um conjunto de informações úteis como o tipo de evento, o tempo de início, o jogador que atuou, a posição, entre outros.

Porém, de acordo com a Wyscout[5], empresa relevante no mercado de produção de dados, a maior parte do trabalho que envolve a produção desses dados é realizada manualmente. Dado o interesse em gerar dados de qualidade sobre os jogos analisados, para cada jogo ao menos 3 analistas especializados em coleta de dados esportivos realizam as anotações dos eventos, sendo que dos 3, um deles é responsável por supervisionar a qualidade dos dados gerados como um todo e os outros 2 são responsáveis por anotar os eventos de cada um dos times em questão. Considerando que grandes campeonatos como a Premier League, liga nacional inglesa, tem 380 jogos por ano e que a produção de dados sobre cada um dos jogos é custosa tanto em termos financeiros quanto em relação ao tempo, a produção desses dados é de alto valor, tornando o acesso à esses dados restrito devido ao seu alto custo final de distribuição.

A automatização da produção de dados estatísticos por meio de análises de vídeos das transmissões dos jogos pode, neste contexto, tornar-se uma ferramenta útil para auxiliar a produção de dados esportivos, principalmente se tratando

da classificação automática do tipo de evento de um dado vídeo. O uso de vídeos para a detecção de ações é uma tarefa abordada por pesquisas recentes, dado que a criação de grandes base de dados de vídeos como Kinetics[4] possibiltou avanços na produção de modelos de reconhecimento de ações que se beneficiam de características extraídas por meio de redes neurais para geração de anotações sobre vídeos. Porém, no contexto de vídeos de futebol a tarefa pode ser desafiadora, dado que a ocorrência de eventos ao longo dos jogos é desbalanceada devido à natureza do esporte, o que faz com que, mesmo que o problema possa ser visto como um subproblema da tarefa de reconhecimento de ações em vídeos, existem particularidades que incentivam a produção de soluções específicas para este tipo de dado.

1.1. Contribuições

Sendo assim, as contribuições deste trabalho se resumem a três fatores principais:

(i) Geração de bancos de dados com vídeos de futebol com eventos devidamente anotados.

(ii) Definição da tarefa de reconhecimento de tipos de eventos em vídeos de futebol e criação de um *baseline* para tal.

(iii) Criação de um arcabouço para replicação dos experimentos realizados e acesso ao código produzido ao longo do projeto, a fim de facilitar contribuições acerca da tarefa definida.

2. Trabalhos relacionados

Este trabalho se relaciona principalmente com os tópicos Reconhecimento de Ações, *Soccer Video Datatasets* e *Soccer Event Datasets*. Faremos, portanto, uma breve análise sobre os principais trabalhos atuais relacionados a cada um desses tópicos.

Reconhecimento de Ações. A tarefa de reconhecimento de ações é caracterizada pela detecção de ações em segmentos de vídeo considerando um leque de classes de ações propostas pelo ser humano. Ao longo do tempo, diversas abordagens foram exploradas para solucionar este tipo de problema, no contexto dos vídeos o direcionamento usual para a solução é a expansão de arquiteturas de redes neurais 2d para o 3d[1], que considera o eixo temporal adicionado pelos vídeos. Desta maneira, os modelos de rede neural gerados são capazes de extrair características tanto espaciais, referentes aos quadros dos vídeos, quanto temporais, relacionando cada quadro levando em consideração o eixo temporal. Um aspecto relevante sobre este tópico é que o custo computacional de grande parte dos modelos que realizam esta abordagem de expansão de redes 2d com arquiteturas tradicionais ao longo do eixo temporal escala rapidamente, tornando os ciclos de estudo demorados.

Outra abordagem possível é o uso de *Transformers* [6] que, no contexto de computação visual, contrasta com os modelos CNN (Rede Neural Convolucional) que focam em extrair características locais, ao propor a extração de características gerais dos dados de entrada, tornando o modelo mais robusto ao lidar com entradas de tamanhos maiores, o que é importante no contexto de dados de vídeo que adicionam o eixo temporal às imagens. Enfim, esses modelos agragam metodologias adotadas por modelos de reconhecimento de imagens, que recorrem à extração de características locais, com características de processamento de linguagem natural ao tentar obter informações utilizando o fator sequencial dos quadros. Como apresentado em [7], as abordagens atuais de classificação visual podem ser categorizadas em 5 tipos principais:

- **Modelos baseados em imagem:** são utilizadas a arquiteturas de extração de características das imagens de cada quadro do vídeo e após a agregação dos vetores de caracteríticas, é utilizado um modelo de classificação tradicional;
- **Redes convolucionais 3D:** são criadas máscaras convolucionais em 3 dimensões que extraem caracteríticas tanto no eixo espacial quanto no eixo temporal simultaneamente, gerando uma nova representação no espaço de características onde de fato o vídeo será classificado;
- **Redes espaço-temporais:** os vídeos são processados separadamente sob dois aspectos, um trata de caracterísicas espaciais e outro das temporais, e uma função de agregação une as informações em uma representação final;
- **Redes neurais recorrentes:** redes que tratam os vídeos considerando o aspecto linear dos frames utilizando de abordagens que usam *LSTM* ou do tipo *Encoder-Decoder* para aprender os atributos que representam os vídeos;
- **Abordagens híbridas:** processos que conciliam aspectos de redes convolucionais e recorrentes para a criação do espaço de representação do vídeos.

Soccer Video Datasets. Tratando da geração de grandes bancos de dados de vídeos de futebol o trabalho que propõe *SoccerNet* [3] ganha destaque. Ao longo do desenvolvimento deste trabalho os autores foram capazes de gerar um banco de dados significativo com imagens de 500 jogos completos das seis principais ligas europeias durante três temporadas, de 2014 a 2017, resultando em um total de 764 horas de vídeo. Além dos vídeos, o banco de dados gerado também contém informações sobre três ações relacionadas aos jogos: gols, cartões e substituições, que foram obtidos extraindo informações dos relatórios dos jogos disponibilizados pelos sites das ligas em questão. Além disso, o trabalho propõe algumas tarefas possibilitadas pelos dados gerados, sendo uma delas a de detecção de ações nos vídeos, que propõe o uso de modelos de classificação de vídeos utilizando segmentos de vídeo gerados por uma janela deslizante de 1 segundo como entrada para gerar anotações sobre os possíveis eventos relacionados ao vídeo,

uma tarefa que se relaciona com uma das propostas deste trabalho que é geração de segmentos de vídeo referentes a eventos de jogos de futebol.

Soccer Event Datasets. Já em relação à geração de dados de eventos associados à jogos de futebol o projeto em enfoque é o que propõe o banco de dados *Wyscout Top 5* [5]. Neste projeto, em associação com a empresa Wyscout, que tem a produção de dados esportivos como seu negócio, os autores desenvolveram um banco de dados que contém dados de eventos de todos os jogos da temporada 2017-2018 das cinco principais ligas da europa (*La Liga, League 1, Premier League, Bundesliga* e *Serie A*), além dos jogos da Eurocopa de 2016 e da Copa do Mundo da Rússia de 2018. Sendo assim, como resultado, tornaram público o acesso a dados de evento que contém informações relevantes sobre cada um dos eventos que ocorreram em cada um dos jogos em questão. Além disso, apresentam a metodologia de coleta desses dados que se apresentava como uma tarefa manual auxiliada por um *software* proprietário da empresa Wyscout. Ao longo do desenvolvimento do projeto há a apresentação de cenários de uso desse tipo de dado que sustentam a contribuição dele com os profissionais de análise esportiva.

3. Métodos

Este projeto propõe a solução de 3 tarefas principais:

- (i) Definição da tarefa de classificação de tipos de eventos em vídeos de futebol e determinação de um *baseline* para tal;
- (ii) Criação de um banco de dados que associe dados de evento à vídeos de futebol;
- (iii) Geração de um arcabouço para replicação dos experimentos e produção de novos testes acerca da tarefa definida.

3.1. Geração do banco de dados

Dado que o objetivo era criar um banco de dados que relacionasse segmentos de vídeos de jogos de futebol referente à eventos com o tipo evento corrente, a estratégia adotada envolve o uso de dados de eventos já existentes para segmentar automaticamente vídeos de jogos completos, de forma a gerar, para cada evento de um jogo completo, um vídeo curto com tipo já referenciado nos dados de evento. Visando este objetivo, três tarefas surgiram neste contexto: coleta dos dados de evento, coleta dos vídeos e segmentação automática dos vídeos.

Coleta de dados de evento: Os dados de evento utilizados para o desenvolvimento do projeto são os do banco de dados *Wyscout Top 5* [5] que foram disponibilizados publicamente pelos autores por meio da plataforma *Figshare*. Ele contém dados de eventos de todos os jogos da temporada 2017-2018 das 5 principais ligas europeias: *La Liga*, *Premier League*, *Bundesliga*, *League 1* e *Serie A*, além dos jogos da Eurocopa de 2016, competição entre seleções europeias, e jogos da Copa do Mundo da Rússia de 2018, maior competição de seleções do esporte. Com isso, estão disponíveis informações sobre 1941 jogos, que no total somam mais de 3 milhões de eventos anotados realizados por quase 4300 jogadores distintos. Sobre cada um dos eventos que ocorreram nesse jogos as seguintes informações estão disponíveis:

1. **eventId:** Um identificador do tipo de ação que ocorre no evento.
2. **eventName:** Nome do tipo de evento que ocorre no evento associado ao eventId.
3. **subEventId:** Um identificador do subtipo de ação que ocorre no evento.
4. **subEventName:** Nome do subtipo de ação que ocorre no evento associado ao subEventId.
5. **tags:** Uma lista que contém informações adicionais sobre o evento, utilizada para especificar alguns tipos de ações que estão subespecificados ainda pelas informações anteriores.
6. **eventSec:** O tempo de início do evento, considerando o tempo em segundos desde o início do tempo atual do jogo.
7. **id:** Um identificador único do evento.
8. **matchId:** Um identificador que referencia o jogo no qual o evento ocorreu.
9. **matchPeriod:** Um identificador do período do jogo em que o evento ocorreu, isto é, identifica se o evento ocorreu no primeiro ou segundo tempo do tempo normal ou prorrogação, ou, por fim, durante os pênaltis.
10. **playerId:** Um identificador do jogador que gerou o evento.
11. **positions:** Uma lista de tuplas contendo as informações sobre as posições de início e término do evento.
12. **teamId:** Um identificador do time do jogador que gerou o evento.

Vale ressaltar que além dos dados relativos aos eventos, o dataset utilizado também contém informações referentes à outros ascpectos acerca dos jogos, como dados sobre os times, os árbitros, os campeonatos, os técnicos, os jogadores e os jogos. Para o escopo de desenvolvimento deste projeto, além dos dados de evento, apenas os dados referentes aos times e jogos foram utilizados para relacionar os jogos coletados aos seus identicadores utilizados no dataset. Além disso, como o objetivo do projeto se tratava da classificação do tipo de evento, apenas os atributos referentes ao tipo de evento, tempo de ocorrência e identificadores tanto do jogo e do evento quanto do período do evento, foram utilizados ao longo do processo de geração do banco de dados.

Coleta dos vídeos: Dado que o interesse era gerar vídeos para os quais já existiam dados de eventos anotados, a coleta dos vídeos foi direcionada à busca por vídeos

Liga	Quantidade de jogos
Copa do Mundo	6
La Liga	42
Premier League	3
Serie A	4
Total	55

Table 1. Esta tabela mostra a distribuição dos vídeos coletados para cada um dos campeonatos dos quais ao menos um vídeo de jogo completo foi coletado.

de jogos completos dos quais havia referência nos dados de eventos coletados, ou seja, dos jogos da temporada 2017-2018 das cinco principais ligas europeias, da Eurocopa de 2016 e da Copa do Mundo da Rússia de 2018. Para isso, foi utilizada a plataforma *Youtube* para realizar a busca, dado que os vídeos listados na mesma são públicos e que os servidores que mantém a aplicação são de grande capacidade, de forma a possibilitar a aquisição de vídeos em tempo hábil para a execução do projeto. Sendo assim, utilizamos os canais oficias das ligas em questão, além dos canais dos times que participam de cada uma dessas ligas, do canal oficial da Fifa e dos canais referentes às seleções que participaram dos campeonatos internacionais, a fim de procurar por vídeos de jogos completos considerando a limitação inicial proporcionada pelo dados de evento. O procedimento utilizado para a procura era simples, por meio da ferramenta de busca interna da plataforma procuramos por vídeos com expressões chaves que remetessem à vídeos de jogos completos como, por exemplo, *'Full Match'* e após realizar as buscas manualmente em cada um dos canais oficiais que teriam potencialmente conteúdos de interesse adquirimos os arquivos referentes aos vídeos por meio de uma ferramenta chamada *Youtube DLP*, uma aplicação pública que possibilita a aquisição de arquivos de diversos sites como o *Youtube*.

Por fim, foi possível coletar vídeos referentes a 55 jogos completos que tem referências aos dados de evento, todos eles com resolução 1920 x 1080 ou 1280 x 720, considerando a limitação dos dados disponíveis e a quantidade de memória que poderia ser alocada para o armazenamento desses arquivos, dado que cada vídeo coletado ocupava entre 1 e 4 Gb de memória devido ao comprimento. Sendo assim, os vídeos coletados são referentes a apenas 4 dos 7 campeonatos dos quais existiam dados de eventos coletados, sendo que grande deles referentes a jogos da *La Liga* como pode ser visto na tabela 1.

Segmentação automática dos vídeos: Para segmentar os vídeos coletados dentre pares de segmentos de vídeos associados à tipos de eventos consideramos dois fatores principais:

- Dentro de um período de um jogo de futebol o tempo é contínuo, ou seja, dado o apito inicial o cronômetro não para até que o período termine;
- Os dados de eventos informam para cada evento o período em que ocorre e tempo decorrido em segundos desde o início do perído até o início daquele evento.

Considerando estes dois aspectos ainda era necessário associar os tempos referentes à um evento com o tempo em que ele de fato ocorre no vídeo. Para isso, foi anotado manualmente para cada um dos 55 jogos coletados os tempos de início para os 2 períodos dos jogos, de forma que, para cada evento, seria possível determinar seu tempo de início no vídeo apenas somando o tempo de início do período correspondente com o tempo de início dado pelos dados de evento, ou seja, dado que *tip* é o tempo de início do período referente àquele evento e *tie* é o tempo de início do evento nos dados de eventos, o *tiv*, tempo de início no vídeo, era $tip + tie$.

Determinado o tempo de início em vídeo ainda era necessário determinar o tempo de fim de cada evento no seu vídeo correspondente. Para isso, duas estratégias principais foram adotadas:

Estratégia 1. Início do evento subsequente: nesta estratégia o tempo de fim de um dado evento no vídeo correspondente era determinado pelo tempo de início em vídeo do evento subsequente, o que foi possível determinar de maneira simples ordenando para cada jogo e período os eventos pelo seu tempo de início. Neste contexto, os últimos eventos correspondentes a cada período de cada jogo apresentavam tempo de fim indeterminado e para fins de aproveitamento de todas as instâncias seus tempos de fim foram determinados pela soma do seus tempo de início e uma constante $c = 1.5$, que foi determinada empíricamente baseada nos tempos médios de duração dos eventos como um todo.

Estratégia 2. Início do evento subsequente ou tempo médio para o tipo de evento: nesta estratégia, o objetivo era tratar eventos que, com a estratégia anterior, apresentaram tempo de duração ($TempoInicio - TempoFim$) incoerentes com o apresentado pelo tipo de evento correspondente. Sendo assim, foram determinados inicialmente tempos de início e fim como na estratégia 1, após isso foi calculado o tempo de duração do evento ($TempoInicio - TempoFim$). Com o tempo de duração determinado foi calculado, para cada tipo de evento, qual a faixa de duração de evento considerada comum para aquele tipo de evento. Para isso, foi calculado para cada tipo de evento os primeiro e terceiro quadrantes da sua distribuição, de forma a determinar um limite superior e inferior para os tempos de duração como $Q1 - 1.5 * IQR$ e $Q3 + 1.5 * IQR$ onde IQR é o intervalo entre o primeiro e teceiro quadrante determinado por $Q3 - Q1$. Desta forma, eventos com tempo de duração fora deste intervalo forma considerados *outliers* e os seus tempos de fim foram determinados de foram alternativa pela soma do tempo de início do evento com a média

do tempo de duração dos eventos daquele tipo.

Determinados os tempo de início e fim para cada evento no vídeo correspondente, a tarefa de segmentar os vídeos era associar esses tempos aos índices de seus quadros no vídeo, utilizando a taxa de quadros(*fps*) do vídeo e os tempos gerados e determinando índices do quadro inicial e final para cada evento como $TempoEmSegundos * fps$. Com isso, foram recuperados os quadros referentes à cada um dos eventos.

3.2. Reconhecimento de eventos

Considerando o objetivo do projeto, a abordagem escolhida foi utilizar uma rede neural convolucional tridimensional para classificação. Neste contexto, a arquitetura escolhida para este processo foi a X3D-S [1] que apresenta uma proposta de expansão de CNN's bidimensionais simples ao longo de eixos dos vídeos como o eixo do tempo e a profundidade dos canais. O motivo pela opção é o custo benefício da proposta que, por expandir redes simples gradualmente pelo eixos do vídeo, geram resultados competitivos com outras abordagens que representam o estado da arte, sem apresentar o custo computacional alto que é um problema recorrente neste contexto. Além disso, dado a quantidade de dados gerados pela etapa anterior e que a arquitetura possui uma grande quantidade de parâmetros, por volta de 3.9 milhões, optou-se pela realização de um *fine-tuning* de um modelo pré-treinado em um banco de dados suficientemente grande para adequar todos os pesos do modelo. Sendo assim, o modelo escolhido para a realização da tarefa foi o modelo pré-treinado no banco de dados Kinetics 400 [4] conhecido no contexto de reconhecimento de ações, que é disponibilizado em uma biblioteca da linguagem *python* chamada *pytorchvideo*.

Dada a natureza do problema a ser resolvido, foi necessário adequar as camadas finais do modelo, que correspondiam ao classificador de fato. Para isso, inicialmente um modelo correspondente à arquitetura proposta foi gerado utilizando a função disponibilizada pela biblioteca *torchvision* e os mesmos hiperparâmetros do modelo pré-treinado. Feito isso, carregamos os pesos referentes às camadas iniciais do modelo que realizam a extração das caracteríticas e não estavam relacionadas diretamente com a tarefa de classificação. Com isso, o modelo gerado para a realização dos experimentos possuía os pesos obtidos com o treinamento no banco de dados *Kinetics* que eram relevantes para a extração de características e pesos aleatórios nas camadas que realizariam de fato a classificação, que foram adaptadas de acordo com o número de classes desejadas na saída para cada um dos experimentos realizados.

4. Experimentos

Os experimentos realizados ao longo do desenvolvimento do projeto visavam entender quais aspectos que poderiam influenciar ao longo da realização do *fine-tuning* do modelo pré-treinado, especialmente acerca de três aspectos principais:

(i) Como a escolha da granularidade escolhida ao definir os tipos de eventos a serem classificados impactam no desempenho do modelo e qualidade dos dados gerados?

(ii) Como a estratégia de segmentação dos vídeos de jogos completos em segmentos já classificados influenciam no resultado gerado?

(iii) Como a quantidade de dados para cada tipo de evento a ser gerado impacta na qualidade da classificação gerada?

Considerando que os modelos seriam gerados a partir de um *fine-tuning* de um modelo de arquitetura *X3D-S* pré-treinado no dataset *Kinetics-400* era necessário utilizar os mesmos parâmetros que geraram estes pesos durante este treinamento ao longo do refinamento. Sendo assim, para cada segmentação gerada pelas estratégias de segmentação 1 e 2, foram gerados tensores de acordo com as dimensões do modelo de (13, 3, 252, 252), onde para cada video foram selecionados com amostragem temporal uniforme 13 quadros e cada quadro apresentava resolução 252x252 e 3 canais, ou seja, considerando um vídeo de evento com 130 segundos, por exemplo, um quadro era selecionado a cada 10 segundos de forma a representar de maneira uniforme o aspecto temporal daquele evento.

Para cada evento presente no banco de dados de eventos referenciados por vídeos foi gerado um tensor com os dados brutos representados em inteiros sem sinal, de forma que para cada pixel era necessário apenas um byte de espaço de armazenamento, de forma que para cada tensor apenas 2.5MB eram necessários para seu armazenamento, devido à limitação do sistema utilizado para a realização dos experimentos.

Para fins de validação e testabilidade, os dados foram separados dentre 3 grupos: treino (60%), validação (20%) e teste (20%), utilizando como referência os jogos, de forma que 33 jogos foram dedicados a treino, 11 à validação e 11 aos testes, a fim de evitar distribuições incoerentes com distribuição real.

Os tensores foram arquivados em um arquivo *h5py* que os organiza de forma eficiente e compacta, tornando possível acessá-los durante o treinamento de forma eficiente por meio de uma chave, que neste contexto foi o identificador único de cada evento. Porém, durante o treinamento foi necessário tranformar os dados para o padrão utilizado durante o treinamento com o dataset *Kinetics-400* que exigia que os números inteiros fossem escalonados no intervalo real [0,1] e fossem normalizados utilizando mesma média e desvio padrão de, respectivamente, 0.45 e 0.225. O processo de processamento dos tensores para o treinamento foi implementado ao definir uma função especializada para função *Dataset* da biblioteca *torch*.

Além disso, todos os treinamentos foram realizados utilizando uma taxa de aprendizado inicial de 0.0001 e otimizador Adam, que utiliza valores em uma janela temporal e valor da função de perda para otimização da taxa de aprendizado ao longo das épocas de treinamento. A validação cruzada foi a função de perda utilizada para os treinamentos, que foram executados por no máximo 20 épocas utilizando de técnica de *early stopping* após 2 épocas consecutivas sem melhora no desempenho considerando a acurácia média entre as classes, ou seja, o modelo era executado sobre todos os dados de validação e era calculado, para cada classe, a sua acurácia, de forma que ao fim fosse possível calcular a média das acurácias que seria utilizada como parâmetro para o fim ou não do processo de treinamento.

Para responder as duas primeiras perguntas de pesquisa 4 experimentos (*exp1 - exp4*) foram realizados, no qual dois atributos foram alternados:

- **Experimento 1:** neste experimento foi utilizada a estretégia de segmentação 1 e o atributo *eventName* para classificar os vídeos dentre as 10 classes possíveis desse atributo.
- **Experimento 2:** neste experimento foi utilizada a estratégia de segmentação 1 e o atributo *subEventName* para classificar os vídeos dentre 36 classes possíveis desse atributo.
- **Experimento 3:** neste experimento foi utilizada a estretégia de segmentação 2 e o atributo *eventName* para classificar os vídeos dentre as 10 classes possíveis desse atributo.
- **Experimento 4:** neste experimento foi utilizada a estratégia de segmentação 2 e o atributo *subEventName* para classificar os vídeos dentre 36 classes possíveis desse atributo.

Já para a última pergunta de pesquisa outros 8 experimentos foram realizados, sendo o experimento 5 uma abordagem de classificação diferente da adotada em geral, que tratou o problema em duas etapas se aproveitando da conexão semântica entre os atributos *eventName* e *subEventName* em que o segundo representa especificações da primeira, o experimento 6 avalia a classificação consirando apenas as 5 classes mais populares e os outros 6 experimentos (*exp7 - exp12*) avaliaram o impacto da quantidade de dados ditribuídos uniformemente na geração do modelo.

No experimento 5, foram utilizados 11 modelos de rede neural, sendo um deles responsável pela classificação primária a fim de determinar o tipo de evento definido por *eventName* e os outros 10 reponsáveis por diferenciar, dentro de cada uma das classes definidas por *eventName* qual o tipo de atributo específico definido por *subEventName*. Enfim, esta abordagem visava gerar modelos com dados mais distribuídos na segunda etapa, uma vez que foi observado que dentro uma classe genérica a distribuição se apresentava mais uniforme do que no escopo geral. Para isso, para a primeira etapa foi utilizado o mesmo modelo gerado no experimento 3, considerando que as configurações se adequavam à este caso. Os outros 10 modelos foram gerados realizando treinamento do modelo apenas com os dados da classe genérica utilizando os rótulos da classe especializada, ou seja, para classe de *eventName 'Pass'* foi gerado um modelo que classificava suas instâncias dentre as suas especializações como, por exemplo, *Simple pass* ou *Smart pass*.

Para o experimento 6, foram filtradas apenas as 5 classes mais populares de *eventName*, e foi realizado um *fine-tuning* do modelo com apenas os dados referentes à estas instâncias, mantendo a ditribuição dos dados originais, de forma apenas a limitar o mínimo de instâncias para cada classe participante.

Para os experimentos 7 a 12, foram filtrados apenas as 5 classes mais populares de *subEventName*, porém utilizando sempre um amostra de tamanho uniforme para cada um das classes, sendo que para cada experimento do 7 ao 12 foram utilizados, respectivamente, os valores 128, 256, 512, 1024, 2048, 4096, ou seja, para o experimento 7, por exemplo, foram selecionadas 128 instâncias para cada uma das 5 classes mais populares para realização do treinamento. Vale mencionar que para a avaliação dos resultados do modelo utilizando o banco de testes foi utilizado a mesma ditribuição dos dados apresentado pelo dado real, de forma a avaliar a qualidade da solução no contexto real e de forma justa se comparada com as outras soluções geradas.

5. Resultados

Para avaliar os resultados dos modelos, considerando as perguntas de pesquisa abordadas pelo projeto, 3 métricas principais foram utilizadas:

- **Acurácia (ACC):** esta medida apresenta a proporção de acertos em relação ao todo, ou seja, dado que 100 instâncias foram avaliadas ao todo e 85 delas foram classificadas corretamente, isto é, a predição do modelo aponta a classe correta com maior probabilidade significa que a acurácia foi de 0.85.
- **Acurácia Média (ACC Média):** esta medida apresenta o equilíbrio na qualidade das classificações para cada classe, ou seja, dada uma classe, é calculada a proporção de acertos em comparação com a quantidade de representantes, e ao fim, calculou-se a média destas acurácias para cada classe.
- **Acurácia Top 3 (ACC Top 3):** esta medida apresenta a porcentagem em relação ao todo de acertos dentre as 3 predições mais prováveis geradas pelo modelo, ou seja, para cada instância foram geradas as predições, os valores foram ordenados de acordo com sua probabilidade e caso a classe correta estive entre as 3 classes mais prováveis ela era considerada um acerto e entrava para a contagem.

Exp	ACC	ACC Média	ACC Top 3
Exp1	0.693	0.438	0.946
Exp2	0.501	0.264	0.749
Exp3	0.692	0.482	0.949
Exp4	0.517	0.277	0.777

Table 2. Esta tabela apresenta os resultados para os experimentos 1 a 4 que avaliam o impacto da variação da quantidade de classes totais e estratégia de segmentação no desempenho final.

Exp	ACC	ACC Média	ACC Top 3
Exp6	0.719	0.637	0.975
Exp7	0.440	0.336	0.813
Exp8	0.500	0.357	0.846
Exp9	0.494	0.397	0.867
Exp10	0.483	0.454	0.820
Exp11	0.522	0.565	0.824
Exp12	0.681	0.754	0.894

Table 3. Esta tabela apresenta os resultados para os experimentos 6 a 12 que avaliam o impacto da quantidade e distribuição dos dados na qualidade do modelo gerado.

5.1. Perguntas 1 e 2

Para as perguntas 1 e 2 era necessário avaliar o impacto da quantidade de classes possíveis e da estretégia de segmentação no modelo gerado, que estão apresentados na tabela 2. Através das métricas é possível notar que não houve diferença significativa de desempenho ao comparar as estretégias de segmentação, ou seja, ao comparar os pares de experimento 1-3 e 2-4, demonstrando que o modelo aparenta ser robusto a ponto de lidar automaticamente com as variações dos dados. Porém, ao avaliar o impacto da quantidade de classes possíveis pelos pares 1-2 e 3-4 há uma diferença significativa entre os desempenhos, demonstrando que o crescimento no número de classes impactou negativamente na qualidade do modelo gerado, principalmente se tratando de Acurácia Média no qual o desempenho para os modelos de 36 classes foi menos de 70% do desempenho do modelo com 10 classes. Porém, é importante denotar que a classificação dos eventos dentre as 36 classes é muito mais valiosa para o fim do dado, que são as análises esportivas, que a classificação dentre 10 classes, pois possibilita a percepção de detalhes em performances que podem ser essenciais no meio profissional.

Enfim, a estretégia de segmentação não aparenta ser fator relevante para a qualidade do modelo. Por outro lado, a medida que a quantidade de classes possíveis cresce a capacidade do modelo em gerar classificações corretas para todas as classes diminui, sendo necessário analisar, para cada classe a ser gerada, o custo benefício de especializá-la ou não dado seu impacto nas análises esportivas.

5.2. Pergunta 3

Para analisar o impacto da quantidade e distribuição dos dados é relevante analisar os resultados dos experimentos 6 a 12 presentes na tabela 3. Neste contexto, é possível notar que à medida que a quantidade de instâncias de cada classe utilizadas para treinamento crescia as métricas em geral melhoravam, principalmente a acurácia média e no top 3, sendo que para os experimentos 6 e 12 que apresentavam a maior quantidade de instâncias avaliadas dentre estes experimentos, os números se demontraram muito superiores se comparados aos treinamentos com números menores de instâncias. Porém, é possível notar que para

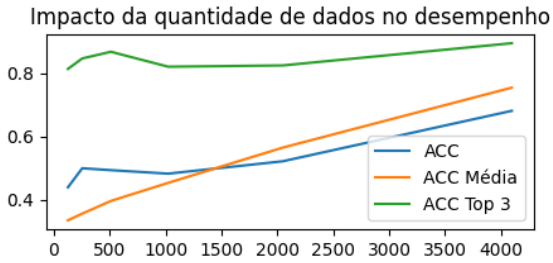

Figure 1. Esta figura mostra a relação entre a quantidade de instâncias para cada classe utilizadas durante o treinamento e o valor gerado para as métricas ao avaliar o modelo correspondente.

as 5 classes mais populares de *eventName* a acurácia geral foi maior (Experimento 6), enquanto para as 5 mais populares em *subEventName* a acurácia média foi maior, o que potencialmente ocorreu porque, apesar das classes geradas serem diferentes, a ditribuição dos dados para o experimento 6 se aproxima mais da distribuição real dos dados, o que pode explicar também o alto valor para a acurácia no top 3, demonstrando que o uso de uma distribuição próxima à real pode impactar positivamente no modelo.

Enfim, é possível notar pela imagem 1 que a quantidade de dados disponíveis para o treinamento é um fator crucial para o seu desempenho final, que tem melhora acompanhada pelo aumento na quantidade de dados. Além disso, a distribuição dos dados, apesar de não ser necessariamente um fator determinante na qualidade do modelo, aparenta ser relevante, sendo que a proximidade da distribuição com a realidade impacta positivamente, no geral, no desempenho.

5.3. Modelo em duas fases

O modelo em duas fases apresentou os seguintes resultados:

- **Acurácia**: 0.501
- **Acurácia Média**: 0.248
- **Acurácia Top 3**: 0.400

A partir das métricas é possível notar, ao comparálas com as métricas dos experimentos compatíves ($exp2$ e $exp4$), que o modelo em duas fases não desempenhou bem, pois mesmo que os valores para acurácia e acurácia

média não fossem significativamente piores, o de acurácia no top 3 foi bem abaixo dos outros experimentos sob mesmas condições. Dentre os principais motivos para o desempenho da abordagem em questão está a propagação de erros da primeira fase para a segunda, que, dado o formato da solução, impossibilita que erros na primeira fase gerem acertos durante a segunda, limitando superiormente os acertos aos acertos da primeira fase e propagando seus erros para a segunda.

5.4. Arcabouço de replicação

Além dos experimentos realizados, um resultado relevante do desenvolvimento do projeto foi a criação de um repositório de códigos e facilitações para a replicação dos experimentos e realização de novos testes ou abordagens acerca da tarefa definida. O repositório em questão está disponível em SoccerEventSpotNet e apresenta 6 módulos principais para a replicação dos experimentos:

- **EventCollector:** responsável pelos códigos utilizados para a coleta dos dados do dataset público da Wyscout, expecificamente os dados de evento e dados sobre times e jogos.
- **VideoCollector:** responsável pelos códigos utilizados para a coleta dos vídeos referenciados pelos dados de eventos, considerando uma lista de links coletados disponibilizada em um módulo auxiliar de nome *Common*.
- **DataProcessing:** responsável pelos códigos necessários para processar todos os dados coletados e prepará-los para o treinamento, desde a filtragem dos dados com referência visual até a geração dos tensores e rótulos para cada experimento.
- **ModelTraining:** responsável pelos códigos que realizam o *fine-tuning* do modelo pré-treinado considerando todos os experimentos realizados.
- **ExperimentsResults:** responsável por manter exemplos de modelos gerados ao fim de cada experimento para fins de praticidade.
- **ExampleNotebooks:** responsável por manter *notebooks python* com códigos de exemplos de processos comuns aos experimentos realizados como forma de facilitar a compreensão acerca do processo adotado como um todo.

6. Limitações

Dados os resultados obtidos através dos 12 experimentos realizados é possível notar que, apesar das soluções geradas apresentarem resultados compatíveis com os apresentados pelo modelo de mesma arquitetura no dataset **Kinetics 400** [1], é possível notar algumas limitações apresentadas ao longo do desenvolvimento do projeto acerca dos seguintes pontos:

(i) Quantidade de dados coletados: é possível notar que, apesar da grande quantidade de dados disponíveis no *dataset* de eventos, foi possível coletar apenas 55 vídeos de jogos completos, que representam uma fração pouco representativa de todos os dados que poderiam potencialmente fazer parte do banco de dados gerados e que, apesar dos mais de 90 mil eventos únicos coletados, podem ter impactado negativamente no desempenho dos modelos.

(ii) Distribuição dos dados: apesar da distribuição apresentada pelos dados ser a distribuição real de um jogo de futebol, a grande disparidade entre a quantidade de representantes para cada uma das classes pode ter sido um fator relevante para os modelos, principalmente pelo fato de que, mesmo com 90 mil instâncias totais no banco de dados, existiam pelo menos 10 classes com menos de 100 representantes.

(iii) Segmentação incoerente: dado que para cada evento existia apenas o tempo de seu início e que os dados de eventos não possuiam marcações de eventos que não fossem referenciados pelas imagens como replays, por exemplo, ambas as estratégias de segmentação podem gerar rótulos incoerentes com o conteúdo do vídeo, que prejudicam o desempenho final dos modelos.

(iv) Arquitetura simples: a arquitetura utilizada, mesmo dentro da família de arquiteturas da qual ela pertence (*X3D*), é simples e com poucos parâmetros se comparados às redes estado da arte, o que as tornam leves porém podem perder capacidade representativa dos vídeos.

7. Trabalhos futuros

Enfim, considerando os resultados obtidos durante o desenvolvimento do projeto, surgem algumas opções de direcionamento para a continuidade do mesmo:

(i): Busca por mais dados de forma a aumentar o número de instâncias de cada classe, principalmente para classes de menor representatividade inerente às características do esporte.

(ii): Uso de modelos com arquiteturas mais robustas com mais parâmetros treináveis a fim de gerar representações mais detalhadas de cada um dos eventos facilitando assim o processo de classificação.

(iii): Uso de heurísticas específicas para segmentação de cada uma das possíveis classes a serem gerados, visando gerar rótulos mais coerentes aos dados visuais e consequentemente aos seus respectivos tensores que servirão de entrada para as redes.

References

[1] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 203–213, 2020. 2, 5, 8

[2] Fifa. Fifa annual report 2024, 2024. 1

[3] Silvio Giancola, Mohieddine Amine, Tarek Dghaily, and Bernard Ghanem. Soccernet: A scalable dataset for action

spotting in soccer videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 1711–1721, 2018. 2

[4] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 2, 5

[5] Luca Pappalardo, Paolo Cintia, Alessio Rossi, Emanuele Massucco, Paolo Ferragina, Dino Pedreschi, and Fosca Giannotti. A public data set of spatio-temporal match events in soccer competitions. *Scientific data*, 6(1):236, 2019. 1, 3

[6] Javier Selva, Anders S Johansen, Sergio Escalera, Kamal Nasrollahi, Thomas B Moeslund, and Albert Clapés. Video transformers: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):12922–12943, 2023. 2

[7] Atiq Ur Rehman, Samir Brahim Belhaouari, Md Alamgir Kabir, and Adnan Khan. On the use of deep learning for video classification. *Applied Sciences*, 13(3):2007, 2023. 2