

UNIVERSIDADE FEDERAL DO PAMPA

Tiago Furtado Drehmer Pinheiro

**Detecção Automática de Eventos em
Partidas de Futebol**

Alegrete
2018

Tiago Furtado Drehmer Pinheiro

Detecção Automática de Eventos em Partidas de Futebol

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação da Universidade Federal do Pampa como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Marcelo Resende Thielo

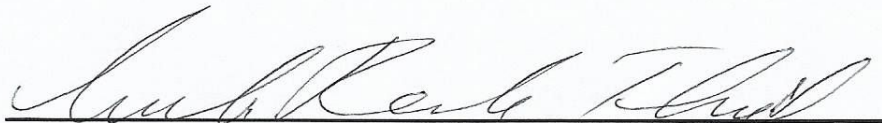
Alegrete
2018

Tiago Furtado Drehmer Pinheiro

Detecção Automática de Eventos em Partidas de Futebol

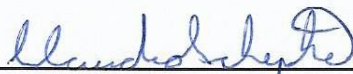
Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação da Universidade Federal do Pampa como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Trabalho de Conclusão de Curso defendido e aprovado em 4 de ~~Dezembro~~ de 2018
Banca examinadora:



Prof. Dr. Marcelo Resende Thielo

Orientador
UNIPAMPA



Prof. Dr. Claudio Schepke

UNIPAMPA



Prof. Me. Diego Kreutz

UNIPAMPA

Aos meus familiares, amigos, colegas e professores.

AGRADECIMENTOS

Gostaria de agradecer principalmente à minha família por me apoiar durante todos esses anos no curso; Minhas amizades, tanto as conquistadas durante os anos de graduação, quanto amizades de longa data, que me ajudaram a superar os desafios que a faculdade trouxe. Um agradecimento a todos os professores, que souberam passar o conteúdo necessário para minha formação como um cientista da computação, e também por me aconselharem durante esse tempo de graduação.

RESUMO

Cada vez mais o mundo dos esportes tem se modernizado através do uso de novas tecnologias, no futebol não é diferente. A tecnologia contribui, por exemplo, no levantamento estatístico das equipes, o que as ajuda melhorar o desempenho tático, assim como o desempenho coletivo e individual dos atletas. O presente trabalho tem como objetivo desenvolver uma ferramenta para a realização do levantamento automatizado dessas informações, através da detecção de eventos em vídeo, nele é utilizado técnicas de aprendizado de máquina e visão computacional. Na área de aprendizado de máquina, técnicas como redes neurais convolucionais e k-means são utilizados para o reconhecimento e classificação de objetos. Na área de visão computacional, é proposto um novo método para a detecção de eventos, baseado em uma máquina de estados definida pelo posicionamento dos jogadores e da bola. Para se obter resultados para a ferramenta proposta, foram feitas diferentes formas de avaliações nela, afim de obter resultados para cada uma das técnicas propostas. Cabe destacar os resultados obtido pelo modelo de rede neural convolucional Faster R-CNN para identificação de objetos e a técnica proposta para classificação dos jogadores, que atingiram uma precisão a cima de 90%. Os resultados obtidos para detecção de eventos variam conforme o evento, para a detecção de passes por exemplo os resultados foram a cima de 90% porém para eventos mais difíceis como roubo de bola a precisão cai para cerca de 70%.

Palavras-chave: Visão Computacional. Detecção de Eventos. Futebol.

ABSTRACT

Increasingly the world of sports has been modernized through the use of new technologies, in football is no different. The technology contributes, for example, to the statistical survey of the teams, which helps them to improve the tactical performance, as well as the collective and individual performance of the athletes. The present work aims to develop a tool to perform the automated collection of this information through the detection of video events, using machine learning techniques and computer vision. In the area of machine learning, techniques such as convolutional neural networks and k-means are used for the recognition and classification of objects. In the area of computer vision, a new method for event detection is proposed, based on a state machine defined by the positioning of the players and the ball. In order to obtain results for the proposed tool, different forms of evaluation were done in order to obtain results for each of the proposed techniques. It is worth noting the results obtained by the Faster R-CNN convolutional neural network model for object identification and the proposed technique for player classification, which reached a precision of over 90 %. The results obtained for event detection vary according to the event, for the detection of passes for example the results were over 90 % but for more difficult events such as stealing the ball accuracy drops to about 70 %.

Key-words: Computer Vision. Event Detection. Soccer.

LISTA DE FIGURAS

Figura 1 – Características de Haar	24
Figura 2 – Método para desenvolvimento do trabalho	31
Figura 3 – Imagens dos jogadores	34
Figura 4 – Região da camiseta jogador	35
Figura 5 – Região da calção jogador	35
Figura 6 – Exemplo da correção altura	37
Figura 7 – Exemplo sem correção da velocidade	38
Figura 8 – Exemplo com correção da velocidade	38
Figura 9 – Estados pertencentes à bola	40
Figura 10 – Sequência de estados que determinam posse de bola	40
Figura 11 – Sequência de estados que determinam passes 1	41
Figura 12 – Sequência de estados que determinam passes 2	41
Figura 13 – Sequência de estados que determinam passes 3	41
Figura 14 – Sequência de estados que determinam um desarme	42
Figura 15 – Sequência de estados que determinam uma jogada individual	42
Figura 16 – Lance passível de falta	43
Figura 17 – Nível de confiabilidade fornecido pelo TensorFlow	46

LISTA DE TABELAS

Tabela 1 – Nivel de relevância dos trabalhos relacionados	32
Tabela 2 – Quantidade de objetos usados para treinamento	33
Tabela 3 – Partidas utilizadas para avaliação	44
Tabela 4 – Resultados das redes neurais convolucionais	45
Tabela 5 – Resultados do modelo Faster R-CNN.	46
Tabela 6 – Resultados da classificação entre times	47
Tabela 7 – Resultados do rastreamento jogadores	47
Tabela 8 – Precisão da definição do domínio	48
Tabela 9 – Resultados obtidos pela detecção de eventos	49
Tabela 10 – Matriz de confusão da detecção de evento	49
Tabela 11 – Resultados obtidos pela detecção de eventos	50

LISTA DE SIGLAS

BSAS Basic Sequential Algorithmic Scheme

COCO Common Objects in Context

FIFA Federação Internacional de Futebol

MCMC Markov chain Monte Carlo

Faster R-CNN *Faster Region - Convolutional Neuron Network*

SSD *Single Shot Multibox Detector*

TCC1 Trabalho de Conclusão de Curso 1

TCC2 Trabalho de Conclusão de Curso 2

SUMÁRIO

1	INTRODUÇÃO	19
1.1	Objetivo Geral	20
1.2	Objetivos Específicos	20
2	FUNDAMENTAÇÃO TEÓRICA	21
2.1	Aprendizado de Máquina	21
2.1.1	Aprendizagem Supervisionada	21
2.1.2	Aprendizagem Não Supervisionada	22
2.2	Visão Computacional	22
2.2.1	Detecção de eventos	23
2.2.2	Identificação de objetos	23
3	TRABALHOS RELACIONADOS	27
4	FERRAMENTAS E MÉTODOS	31
4.1	Análise de trabalhos relacionados	31
4.2	Metodologia para identificação dos objetos	33
4.3	Metodologia para classificação dos jogadores Entre os times .	34
4.4	Metodologia para rastreamento e suavização dos jogadores entre os frames	36
4.5	Metodologia para detecção de eventos	38
4.5.1	Posse de bola entre as equipe	39
4.5.2	Passes certos, errados e interceptações de cada equipe	40
4.5.3	Desarmes e Jogadas individuais	42
4.5.4	Disputa de Bola	43
4.6	Ferramentas para implementação da aplicação	43
4.7	Avaliação dos resultados	44
5	RESULTADOS OBTIDOS	45
5.1	Identificação de objetos	45
5.2	Classificação dos jogadores em times	46
5.3	Rastreamento dos jogadores	47
5.4	Definição dos estados e domínio	47
5.5	Detecção de eventos	48
5.6	Detecção de lances possíveis de faltas	49
6	CONCLUSÕES	51
	REFERÊNCIAS	53

Índice	55
------------------	----

1 INTRODUÇÃO

O futebol é um dos esportes mais populares do mundo, isso ficou evidente na copa do mundo de 2014, quando o número de espectadores da partida final superou os 3 bilhões, segundo dados fornecidos pela Federação Internacional de Futebol (FIFA)¹. O alto interesse das pessoas por este esporte exige um investimento cada vez maior em tecnologias, que tem como objetivo o aprimoramento, tanto das equipes quanto dos atletas individualmente. Entre as tecnologias, os softwares de análise têm se tornado essenciais. Tais softwares utilizam de um grande volume de dados, que indicam como melhorias podem ser alcançadas. Atualmente o levantamento desta grande quantidade de dados não é totalmente automatizado, havendo ainda a necessidade de intervenção humana para complementá-lo. Em vista disto, este trabalho tem como objetivo definir uma nova forma de como o levantamento de alguns desses dados pode ser automatizado.

Segundo Khan et al. (2018) os dados tipicamente coletados de um jogo de futebol incluem: gols marcados, assistências, número de chutes a gol, informações sobre posse, escanteios, laterais, faltas, cartões dados, lesões, substituições, etc. Essas informações são os fatores que influenciam nos resultados das partidas, e é por meio delas que o futebol evolui. A grande quantidade de vídeos fornecidos do esporte permite que essas informações possam ser coletadas através de técnicas de visão computacional. A visão computacional é capaz de identificar diferentes tipos de eventos presentes nas partidas e, através disso, torna-se possível então a automatização do levantamento de dados para utilização em softwares de análise.

Técnicas de visão computacional como identificação de objeto e detecção de eventos são utilizadas neste trabalho. A identificação de objetos foi determinada pela utilização de métodos de aprendizado de máquina, onde foi possível fazer o reconhecimento e classificação do que é relevante para o desenvolvimento da aplicação. Foi proposta uma forma de rastreamento dos objetos ao longo dos *frames* e uma forma para classificação dos jogadores em diferentes times. Na detecção de eventos é proposta uma forma de reconhecimento de eventos nas partidas, definida por uma máquina de estados onde cada estado é representado pelo posicionamento da bola e jogadores nas imagens. Os eventos então são caracterizados pela sequência destes estados.

Este trabalho consiste nas seguintes partes: Inicialmente, é apresentada uma fundamentação teórica sobre os principais temas abordados, seguido de uma breve explanação do mecanismo dos trabalhos relacionados já publicados, logo após é apresentada a metodologia de como a ferramenta foi desenvolvida e, por fim, os resultados obtidos e uma conclusão sobre os eles

¹ Disponível em: <<https://www.fifa.com/>>.

1.1 Objetivo Geral

Desenvolvimento, utilizando técnicas de visão computacional e aprendizado de máquina, de uma aplicação para reconhecimento de diferentes tipos de eventos que ocorram em partidas de futebol como, por exemplo, posse de bola, passes e chutes a gol.

1.2 Objetivos Específicos

- Fazer o reconhecimento dos jogadores e da bola em vídeo de uma partida de futebol.
- Fazer o rastreamento dos jogadores e da bola ao longo dos *frames* dos vídeos.
- Fazer o reconhecimento de diferentes tipos de eventos.

2 FUNDAMENTAÇÃO TEÓRICA

A seção a seguir apresenta o conteúdo necessário para o desenvolvimento da ferramenta proposta. São apresentadas informações sobre as áreas exploradas, assim como as técnicas pertencentes a elas, que auxiliaram no desenvolvimento da ferramenta.

2.1 Aprendizado de Máquina

Aprendizado de Máquina é uma área da inteligência artificial que estuda e desenvolve a capacidade dos computadores de aprenderem por conta própria o que um conjunto de dados de um universo pode significar. Para Monard e Baranauskas (2003) o aprendizado de máquina é responsável pelo desenvolvimento de técnicas computacionais para a construção de sistemas que são capazes de adquirir conhecimento de forma autônoma. Segundo Goldschmidt (2010), "O Aprendizado de Máquina é uma área da Inteligência Computacional que investiga meios para tornar computadores capazes de aprender a partir da experiência". A principal característica no aprendizado de máquina é o aprendizado indutivo, que representa a capacidade que um organismo tem de raciocinar sobre exemplos fornecidos por processos externos ao sistema de aprendizado (MONARD; BARANAUSKAS, 2003). Esse aprendizado pode ser dividido em supervisionado e não supervisionado. Na sequência são apresentados as definições e como podem ser aplicados estes métodos de aprendizado:

2.1.1 Aprendizagem Supervisionada

Na aprendizagem supervisionada é fornecido ao indutor um conjunto de exemplos de dados para o treinamento, junto a esses exemplos é fornecido a classe que define o que aquele conjunto de dados representa. (MONARD; BARANAUSKAS, 2003). Nesta forma de aprendizado, os exemplos de dados então devem carregar a informação esperada a ser obtida pelo modelo de conhecimento que está sendo construído (GOLDSCHMIDT, 2010). Neste modelo de aprendizado então, os resultados esperados para as classes são fornecidos durante o treinamento e, após isso, o computador é capaz de prever os resultados em relação aos dados de entrada que forem utilizados. Dos métodos de aprendizagem supervisionada para o trabalho foi utilizado o de redes neurais artificiais, que é uma tecnologia que apresentou bons resultados para identificação de objetos.

A primeira aparição de uma rede neural artificial foi proposta por McCulloch Warren S e Pitts (1943), onde foi apresentado um modelo de neurônio artificial que possui dois estados, ligado e desligado. O estado de cada neurônio depende dos estímulos que recebe considerando o comportamento dos neurônios aos quais está conectado. No trabalho se definiu também, que esses modelos poderiam aprender sozinho. Para desenvolvimento da ferramenta aqui proposta o modelo de redes neurais escolhido, foi de redes neurais convolucionais, as quais foram utilizadas na identificação de objetos. Uma melhor explicação

sobre esse modelo de rede neural artificial pode ser encontrado na subseção 2.2.2.

2.1.2 Aprendizagem Não Supervisionada

Nesta forma de aprendizado os algoritmos usam a similaridade que o conjunto de dados apresenta para agrupar os exemplos de dados, de forma que os mais parecidos fiquem no mesmo grupo (GOLDSCHMIDT, 2010). Assim sendo, os exemplos utilizados para treinamento não contém a respeito do que eles representam, deixando essa definição para a máquina. Segundo Shalev-Shwartz e Ben-David (2014), no aprendizado, não supervisionado não existe diferença entre os dados de treinamento e de teste, e a cada nova entrada o sistema se reajusta para melhor representar as classes. No presente trabalho, como método de aprendizado de máquina não supervisionado, foi utilizado o algoritmo *k-means*.

O *k-means* então pode ser usado para agrupamento de cores semelhantes, dando uma sequência de cores definidas por valores RGB. Ele faz com que as que apresentem valores próximos fiquem no mesmo grupo. Este método foi proposto por Lloyd (1982) e é utilizado para agrupamento de informações em k grupos definidos por um conjunto de centroides. O melhor agrupamento é definido pela menor distância dos valores pertencentes aos indivíduos em relação ao seu centroide. O valor de um centroide é definido pela média dos valores das informações que pertencem ao seu grupo.

2.2 Visão Computacional

Visão computacional é uma área da ciência da computação responsável por interpretar dados, processá-los e obter resultados em relação às imagens capturadas por qualquer que seja o dispositivo. Ela representa uma serie de técnicas que são utilizados para exploração científica e técnica. (JAHNE, 2000). Tendo como objetivo descrever o mundo que vemos em uma ou mais imagens e reconstruir suas propriedades, como curvas, iluminação e distribuição das cores (SZELISKI, 2010). Conforme Parker (2010), o objetivo da pesquisa em visão computacional é fornecer aos computadores recursos de percepção semelhantes aos de seres humanos para que eles possam compreender o ambiente, entender os dados observados e tomar decisões apropriadas, assim como aprender com essas experiências e melhorar seu desempenho futuro.

Segundo Moeslund, Thomas e Hilton (2014), o mundo dos esportes envolve movimentos rápidos e precisos que são desafiadores para os competidores, mas que também são para treinadores e espectadores que desejam acompanhar e analisar as partidas. Sendo assim a visão computacional tem muito a acrescentar, já que através da utilização de seus métodos é possível fazer o levantamento de informações de maneira automática, através de vídeos. Dados que indicam como equipes e jogadores se comportam, e definem como o desempenho deles podem ser melhorados. Além disso auxilia os interessados a

compreender o que está acontecendo no esporte de uma maneira mais fácil e objetiva.

2.2.1 Detecção de eventos

Como uma subárea da visão computacional, a detecção de eventos, tem como objetivo identificar diferentes acontecimentos presentes em vídeos. Segundo Ke, Sukthankar e Hebert (2007), "O objetivo da detecção de eventos é identificar e localizar padrões espaço-temporais especificados em vídeo". No futebol, a detecção de eventos é dividida entre as interações de jogadores com a bola e jogadores com outros jogadores (YU et al., 2003), e uma estratégia para o reconhecimento dessas interações é a divisão dos eventos complexos em eventos mais simples. De acordo com Khan et al. (2018), os passos para a detecção desses eventos mais complexos envolvem a detecção de eventos de baixo nível, para então em seguida, através da composição desses, obter eventos de mais alto nível. Eventos simples, seriam como características presentes em eventos mais complexos; por exemplo, um chute a gol que é caracterizado pela sequência de um chute e a trajetória da bola conectando a posição do chute com o local do gol.

2.2.2 Identificação de objetos

A identificação de objeto é responsável por descobrir e identificar a existência de objetos em uma área pertencente a uma imagem (SOO, 2014). Os métodos aplicados nesta área podem ser aplicados através de uma varredura nos pixels a fim de encontrar o objeto, ou através do reconhecimento de características para o objeto que deseja identificar. Essas duas abordagens se diferenciam principalmente pela quantidade de processamento, pois o número de características é substancialmente inferior em relação ao número de pixel de um objeto. (MOREIRA, 2008). Os atuais sistemas de detecção de objetos, então, procuram seguir a seguinte abordagem: criar hipóteses de caixas delimitadoras, redimensionar pixels ou recursos para as caixas e aplicar um classificador (LIU et al., 2016). A seguir são apresentados alguns métodos para o reconhecimento de objetos em imagens.

- Cores de pixel: A técnica mais antiga para identificação de objetos é através das cores de pixel, onde é percorrida a imagem procurando cores de pixels que podem caracterizar um objeto. Para isso então, são utilizados cálculos matemáticos que determinam a presença do objeto ou não em uma imagem.
- Cascata de Haar: Na cascata de Haar primeiro se extraem as características, que são baseadas no filtro Haar e foram definidas por Viola e Jones (2001), possuindo três formatos: 1 - Característica de "dois retângulos", onde o seu valor é a diferença da soma dos pixels pertencentes às duas regiões retangulares de mesmo tamanho; 2 - Característica de "três retângulos" cujo o valor é definido pelo valor de um retângulo central menos a soma de dois retângulos externos; 3 - Característica de "quatro

retângulos” que o valor consiste na diferença entre pares diagonais de retângulo. Essas características são apresentadas na Figura 1 e são extraídas pelo algoritmo Adaboost (FREUND; SCHAPIRE, 1997). Com a extração pronta, aplica-se um classificador em cascata, no qual é utilizada uma adaptação do algoritmo AdaBoost sobre exemplos de imagens negativas e positivas do objeto que se deseja identificar, para a definição das funções das camadas do classificador. Este modelo então faz com que possíveis regiões onde possa existir o objeto, sejam eliminadas por suas camadas, e caso essa região passe por todas elas, significa que objeto esta ali representado. (MOREIRA, 2008)

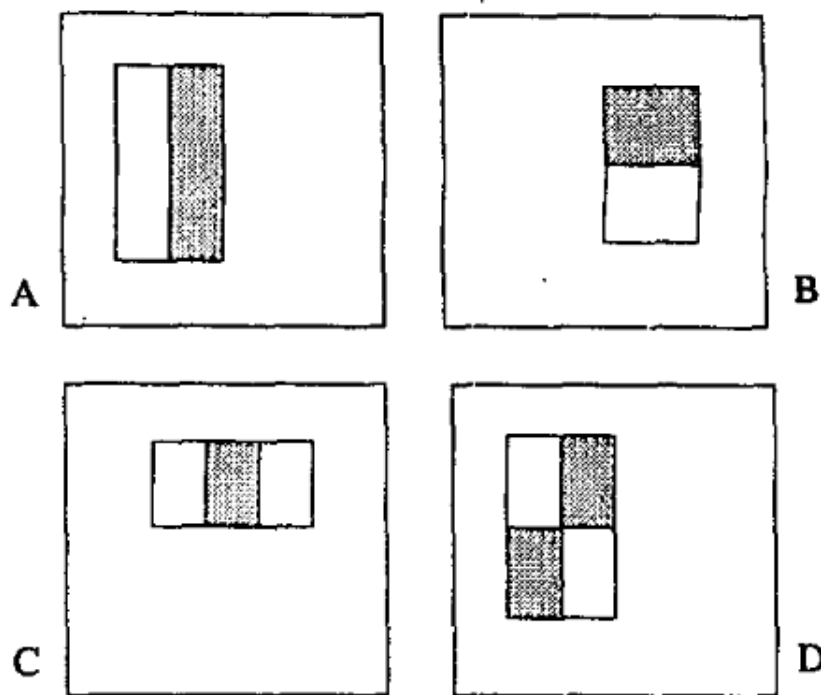


Figura 1 – Características de Haar

Fonte: Viola e Jones (2001)

- Rede Neural Convolutacional: Técnica de aprendizado de máquina que vem sendo usada na área de detecção de objetos. Essa técnica consiste em um tipo de rede neural multicamada que, assim como outras redes neurais, são treinadas com uma versão do algoritmo *backpropagation* (LECUN et al., 2015). A ideia para a utilização dessas redes em reconhecimento de objetos é inspirada em um processo biológico, onde as conexões entre os neurônios presentes na rede são inspiradas no córtex visual identificado nos animais (MATSUGU et al., 2003). Neurônios corticais individuais são estimulados em regiões restritas do campo de visão, as quais são denominadas campos receptivos, esses campos de diferentes neurônios se sobrepõem parcialmente de maneira a cobrir todo o campo de visão. Segundo (GIRSHICK et al., 2014),

a identificação de objeto utilizando dessa técnica consiste em três módulos: gerar regiões candidatas para o detector, extração de características através de uma rede neural convolucional, e utilização de uma máquina de vetores de suporte para especificação de classes.

A identificação de objetos através das cores de pixels é o modelo mais antigo dos três, e menos usado nas tecnologias de hoje em dia, pelo alto nível de processamento necessário e alta complexidade para implementação. Tanto a cascata de Haar quanto as redes neurais convolucionais, foram um salto para o método em suas épocas. A cascata de Haar, surgida em 2001, depende da frequência das características de Haar para apresentar um bom resultado. Redes neurais convolucionais são um método mais atual e mais utilizado nessa área, sendo essa uma técnica em rápida evolução e com metodologias que já demonstram ganho em sua aplicação, como é caso do modelo *Faster Region - Convolutional Neuron Network (Faster R-CNN)* (REN et al., 2015), e o modelo *Single Shot Multibox Detector (SSD)* (LIU et al., 2016).

3 TRABALHOS RELACIONADOS

Nesta seção é apresentado o processo de revisão bibliográfica para obter os trabalhos relacionados. O processo consistiu em três etapas: pesquisa, filtragem e avaliação de trabalhos. A etapa de pesquisa consistiu na pesquisa por palavras-chave que definem este trabalho, como visão computacional, futebol, detecção de eventos, etc. Foi utilizada a ferramenta de pesquisa *google* acadêmico pois nela se concentram resultados de diferentes bases de dados como IEEE (Institute of Electrical and Electronics Engineers)¹, Springer², ACM (Association Computing Machinery)³, dentre outras.

Com a conclusão da primeira etapa havia a necessidade de uma filtragem dos resultados obtidos. A filtragem consistiu de, em uma primeira fase, ler os títulos e resumos dos artigos e, a segunda fase, ler a introdução e resultados. As duas fases tinham como objetivo descartar os trabalhos que apresentassem menos relevância. Por fim os trabalhos que restaram foram para a última etapa.

Na última etapa foi feita uma avaliação sobre os trabalhos encontrados na primeira etapa e aprovados pela segunda. Essa avaliação foi feita sobre o conteúdo geral do trabalho, verificando o que foi desenvolvido e como, sendo assim os trabalhos com avaliação melhor foram aceitos como auxílio para elaboração deste trabalho. A seguir podemos ver o resultados obtidos após a conclusão das três etapas.

No trabalho Khan et al. (2018), foi desenvolvida uma aplicação capaz de detectar eventos comuns presentes em uma partida de futebol, como por exemplo a posse de bola por um jogador e chutes. A aplicação apresentada é capaz de realizar a detecção da bola e dos jogadores da partida, sendo capaz de classificá-los entre os diferentes times. Essa detecção é realizada através do modelo SSD que, ao contrário de outras propostas para esse fim, une as tarefas de selecionar uma região de um possível objeto à de classificação dele, através de um filtro convolucional que mapeia as características pertencentes à área identificada. Após a identificação dos objetos ele desenvolveu uma forma de detecção de eventos, onde dividiu os mesmos em simples e complexos. Eventos simples consistem na relação entre os objetos e eventos complexos são a combinação de outros eventos pré-definidos. Na área de identificação de objetos o algoritmo teve um resultado de 77% para reconhecimento da bola e 90% para identificação dos jogadores. Os eventos detectados no trabalho desenvolvido por ele, foram o de condução da bola e chute, onde o chute é definido pelo afastamento da bola de um jogador, e o domínio é representado pela proximidade da bola a um jogador durante um período de tempo. A precisão do algoritmo foi de 92% para posse de bola e 84% para chutes. Finalmente, em seu trabalho é proposto como trabalho futuro a detecção de outros eventos como o chute a gol ou passe entre jogadores.

A aplicação desenvolvida por Liu et al. (2009) tem as funcionalidades de detecção, classificação e rastreamento de jogadores em uma transmissão de futebol. O método

¹ Disponível em: <<https://ieeexplore.ieee.org/Xplore/home.jsp>>.

² Disponível em: <<https://www.springer.com/br/>>.

³ Disponível em: <<https://dl.acm.org/>>.

proposto inicialmente extrai a cor dominante do vídeo acumulando histogramas de cores HSV. Após a extração da cor dominante, ele então usa cascata de Haar, utilizando exemplos de imagens onde parte delas representa regiões onde os jogadores estão presente, para o treinamento do modelo, assim foi possível identificar os atletas e o juiz presente na partida. Feito isso, havia a necessidade de classificá-los em diferentes times, e a técnica usada foi a extração da cor dominante, presentes nos exemplos recolhidos pelo identificador dos atletas, e a remoção da parte inferior da imagem, mantendo só a parte de cima (uniformes) dos praticantes. Através dos pixels restantes foram identificadas as cores dominantes nos uniformes dos mesmos, permitindo assim a classificação dos jogadores e do juiz presentes no vídeo. Por fim, o autor adicionou em sua aplicação uma forma de rastreamento dos jogadores, onde usou se uma representação em grafos para definir trajetórias dos jogadores em diferentes *frames* e, através do algoritmo Markov chain Monte Carlo (MCMC), se fez uma distribuição de probabilidade para calcular uma solução ótima. Como resultado para identificação de jogador e classificação, o trabalho mostra uma taxa de acerto de aproximadamente 90,00%, já para o rastreamento do jogador foi obtido 98,45% de sucesso.

No artigo publicado por Yu et al. (2003), é proposto o rastreamento da bola em uma transmissão de uma partida de futebol. A ferramenta apresentada é dividida em fases, sendo que a primeira fase consiste em estimar o possível tamanho da bola, a qual é feita através da comparação com pontos de referência no gramado. A segunda fase consiste em detectar os objetos no vídeo que possam ser identificados como a bola. Para isso, ele conta com a identificação de algumas informações, como o tamanho que foi estimado, uma média das cores de pixels pré-definidas presentes na bola e a eliminação de objetos com geometria diferente de um círculo ou elipse. Com essas informações, é calculada uma probabilidade para as possíveis bolas em cada *frame*. A ferramenta então determina trajetetos entre as candidatas à bola, nos *frames* em sequência. Esses trajetetos então são estendidos de maneira a cobrir obstruções da imagem da bola, como no caso dela se posicionar atrás de um jogador ou sair fora do campo de visão da câmera. Então, a trajetória que possui maior valor para somatório das probabilidades indicadas na primeira fase, é escolhida como a correta. Após o reconhecimento da trajetória da bola foram desenvolvidas no trabalho algumas formas de detecção de eventos, sendo os mais relevantes o toque em bola e o evento de posse de bola. Para detecção do toque em bola utilizado o cálculo da velocidade da bola, definindo como um toque então, a alteração nessa velocidade. Para determinar a posse de bola por time, ele propõe a identificação dos jogadores verificando a vizinhança da bola, e extraíndo as informações do que seria o jogador. Em seus resultados é mostrado que sua proposta para detecção de eventos, teve uma precisão de 95,6% de corretude para o toque na bola, 100% para passes e gols, 82% bolas perdidas pelos jogadores, 83,38% para posse de bola e 96,15% de precisão para detecção da bola em jogo ou não.

No trabalho feito por Spagnolo et al. (2007), foi desenvolvida uma ferramenta capaz de identificar os jogadores e distingui-los entre os times. Sua aplicação conta com a extração do *background*, que é feita calculando a energia de um ponto da imagem dentro de um intervalo de tempo, onde pontos com uma grande energia significam pontos que se movem, já pontos com uma energia mais baixa, são pontos pertencentes ao *background*. Com a extração do *background* foi possível extrair regiões que não pertencem a ele, ou seja, os jogadores, juiz e a bola. Essas regiões ainda precisavam ser classificadas - para isso foram definidas características, nas quais foram usados histogramas de cores RGB pertencentes aos exemplos recolhidos, visto que são as cores das camisetas que distinguem os exemplos. Na classificação então foi usada uma variação do algoritmo Basic Sequential Algorithmic Scheme (BSAS). Nesse algoritmo é definido um número máximo de *clusters* sendo que, em seu treinamento, ele pode detectar menos que esse número. O algoritmo foi treinado usando como distância entre os pontos o cálculo da distância de Manhattan. Para testar o algoritmo, foi utilizada uma câmera fixa apontada para o meio do campo da partida, e foram utilizados vídeos de cinco partidas, onde a avaliação do classificador foi dividida entre os dois períodos, obtendo assim 10 resultados. O tempo de processamento teve uma média de 109,6 ms, e dos 10 testes 9 conseguiram detectar o número máximo de *clusters*, além disso ele apresenta o grau de confiança do algoritmo onde se obteve aproximadamente 93,18 % de acerto na classificação em um dos testes essa precisão foi reduzida pelo fato de um dos uniformes do time ser parecido com o do juiz, fazendo com que o algoritmo errasse a classificação.

No trabalho desenvolvido por Tovinkere e Qian (2001), foi proposta uma técnica para detecção dos eventos semânticos presentes em um jogo de futebol, por meio do rastreamento dos atletas e da bola na partida. A técnica reconhece diferentes ações presentes no jogo, desde ações mais simples como uma corrida até mais complexas como a defesa de um chute feita por um goleiro. Nele os autores ainda dividem os eventos em observáveis e interpretados, onde eventos observáveis são aqueles possíveis de ser identificados apenas pela amostra de vídeo, como por exemplo um chute ou um gol, enquanto eventos interpretados são decorrentes da interpretação dos eventos observados como um chute a gol. A metodologia usada para o reconhecimento dos eventos em seu trabalho é feita por uma heurística de um modelo hierárquico, capaz de identificar as ações entre entidades, as quais consistem em conhecimentos prévios de eventos no futebol. Para a validação do trabalho foi implementado um algoritmo onde as entradas eram o posicionamento dos jogadores e da bola, e com isso foi possível detectar os eventos de chute, defesa, chute a gol, desvio na trajetória da bola entre outros, a média dos resultados ficou com 97% de acerto.

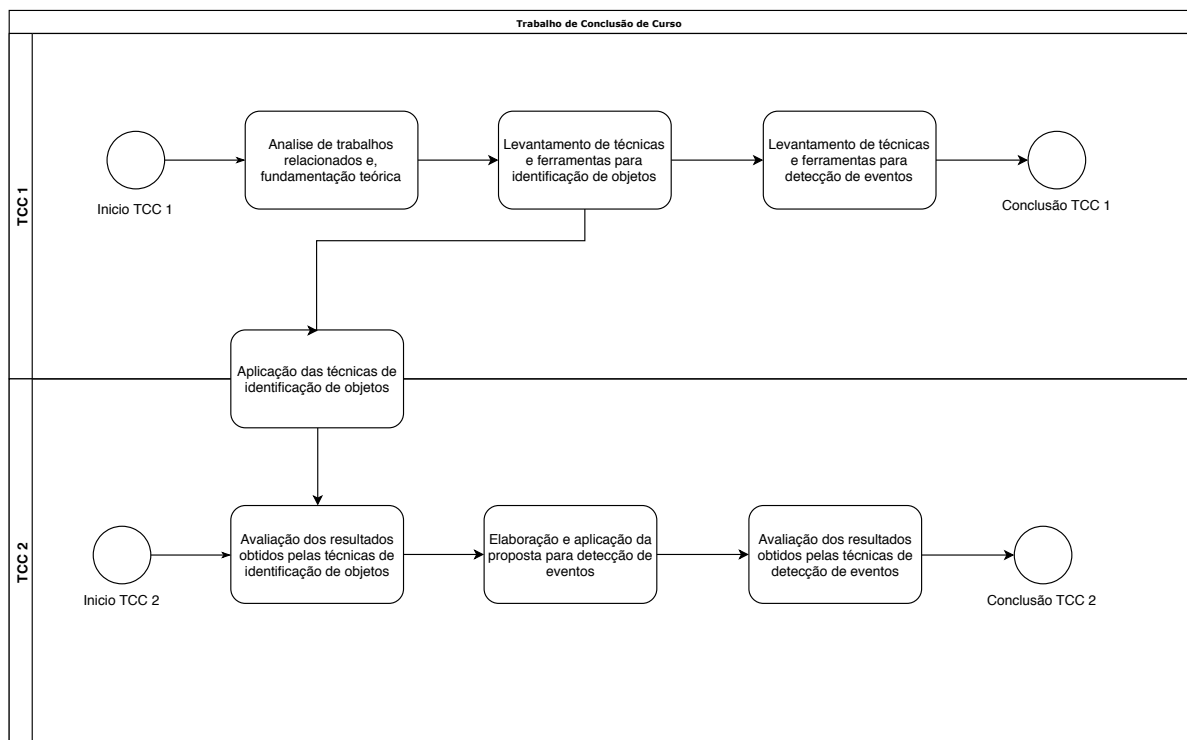
No trabalho desenvolvido por Tavassolipour, Karimian e Kasaei (2014), ao contrário das outras presentes nesta seção, sua contribuição não é uma detecção de eventos em relação à interação entre os objetos presentes na partidas. Neste caso específico, os

autores focam em resumir eventos presentes em uma transmissão de futebol, caracterizando os eventos através do reconhecimento do que está sendo transmitido para os telespectadores. Para tanto, ele utiliza o reconhecimento de *replays*, identificação do logo da transmissão, dos diferentes tipos de filmagem presentes na transmissão, além de outros aspectos presentes apenas em transmissões de futebol. Os eventos detectados nesse trabalho são: gols, cartões amarelos, tentativas de gols, escanteios, faltas, impedimentos entre outros e são utilizados para aplicação de um resumo para a partida.

4 FERRAMENTAS E MÉTODOS

O método desenvolvido para a elaboração deste trabalho está representado na Figura 2, sendo dividido em quatro etapas para o Trabalho de Conclusão de Curso 1 (TCC1) e outras quatro para o Trabalho de Conclusão de Curso 2 (TCC2). A etapa "aplicação das técnicas de identificação de objetos" foi desenvolvida em ambas as *lanes*, enquanto a etapa de "levantamento das técnicas para identificação de objetos", pertencente ao TCC1, determinou como seriam as etapas de aplicação e avaliação da identificação de objetos, definidas para o TCC2.

Figura 2 – Método para desenvolvimento do trabalho



4.1 Análise de trabalhos relacionados

Nesta seção é apresentada uma análise sobre os trabalhos relacionados. A análise consistiu em destacar os métodos neles aplicados, assim como a definição de um grau de relevância que o trabalho tem para o desenvolvimento deste. Na Tabela 1 são apresentados cada um dos trabalhos relacionados junto a sua avaliação de relevância.

O primeiro trabalho indica uma forma de reconhecimento dos jogadores e da bola em uma partida, através do uso do algoritmo SSD. No trabalho também é proposta uma forma para a detecção de eventos classificando-os em baixo nível e alto nível, sendo

Tabela 1 – Nivel de relevância dos trabalhos relacionados

Trabalho	Relevância
Soccer Event Detection	Alta
Automatic player detection, labeling and tracking in broadcast soccer video	Alta
Detecting Semantic Events in Soccer Games: Towards A Complete Solution	Alta
Trajectory-Based Ball Detection and Tracking with Applications to Semantic Analysis of Broadcast Soccer Video	Média
An Unsupervised Approach for Segmentation and Clustering of Soccer Players	Baixa
Event Detection and Summarization in Soccer Videos Using Bayesian Network and Copula	Baixa

que eventos de baixo nível são determinantes para a detecção de eventos de alto nível. Podemos concluir ainda, que os princípios de alguns eventos detectados no trabalho vieram a ser úteis para o desenvolvimento deste, como o evento de posse de bola. O trabalho tem alta relevância pelo alto grau de similaridade com o que estamos propondo.

O segundo trabalho desenvolveu a identificação de objetos, usando o modelo Cascata de Haar, que é apresentado na subseção 2.2.2. Além da detecção de objetos, também é apresentada uma forma de rastreamento dos jogadores entre os *frames*. Para isso foi utilizado o algoritmo MCMC. A relevância desse trabalho é alta por apresentar uma forma de detecção de objeto e, também, o desenvolvimento de uma forma de rastreamento dos jogadores, que foram princípios utilizados no desenvolvimento de nossos métodos.

O terceiro trabalho apresenta uma proposta para detecção de evento, para isso, o modelo recebe como entrada o posicionamento dos jogadores e da bola ao longo do tempo. Ele utiliza de métodos matemáticos para calcular eventos simples, como o desvio na trajetória da bola. Esse trabalho possui alta relevância por elaborar uma forma de detecção de eventos através do posicionamento dos jogadores e bola.

O quarto trabalho, apesar de ser mais antigo, tendo sido publicado há mais de quinze anos, possui média relevância, pois sugere uma forma de rastreamento da bola, assim como a detecção de alguns eventos como um toque na bola, evento que é verificado através da detecção de alteração de velocidade desta.

No quinto trabalho é apresentada uma forma de extração do *background*, sobrando assim áreas onde os jogadores estão presentes, os quais são então classificados, utilizando uma variação do algoritmo BSAS. Esse trabalho possui relevância por definir uma forma de classificação dos jogadores através da cor da camiseta, o que vem a ser útil para a divisão dos times presentes na partida.

O sexto trabalho é relacionado à detecção de eventos através de uma rede neural Bayesiana, aonde as *features* são extraídas de acordo com métodos propostos no trabalho como detecção de quebra entre os diferentes tipos de filmagens presentes na transmissão, classificação do tipo de filmagem utilizada e detecção de *replays*. Esse trabalho não possui muita relevância por se basear em características de transmissões de televisão para fazer

a detecção de eventos.

4.2 Metodologia para identificação dos objetos

Para a identificação dos objetos inicialmente foi realizado um levantamento das tecnologias já existentes na área. Este levantamento resultou em duas abordagens de interesse, sendo elas o método da cascata de Haar e das redes neurais convolucionais. Cascata de Haar (VIOLA; JONES, 2001), sendo uma tecnologia mais antiga, possui certas limitações, por exemplo apresentando uma baixa quantidade de características para reconhecimento, o que a torna inviável para a utilização neste trabalho. Redes Neurais Convolucionais é uma tecnologia mais atual possuindo diferentes maneiras de serem implementadas, as quais acabam interferindo na qualidade do resultado final. No trabalho foram utilizados dois desses métodos, sendo eles: SSD e *Faster* R-CNN.

Essas duas redes neurais apresentam um balanceamento entre a velocidade de reconhecimento e precisão dos acertos. Enquanto SSD apresenta uma velocidade maior, podemos dizer que ela apresenta menor taxa de sucesso para o reconhecimento dos objetos na imagem. Para definir qual das redes atendeu melhor às necessidades do trabalho, foi definido que uma melhor qualidade teria maior relevância do que o tempo de processamento, já que é uma ferramenta para a qual se objetiva o uso após uma partida (*offline*), e não durante (*online*), ou seja, com tempo de folga para o processamento. A ferramenta *TensorFlow* já fornece esses dois modelos de rede para a identificação de objetos, treinadas a partir da base dados Common Objects in Context (COCO). Esses modelos pré-treinados não apresentaram bons resultados, o que fez com que houvesse a necessidade de se realizar um novo treinamento.

Para o treinamento das redes, separou-se *frames* de cada um dos vídeos que iria ser aplicado o algoritmo e, a partir deles, foi feito uma rotulamento das imagens, definindo qual região representava os objetos. Na Figura 3 estão representadas alguma dessas regiões que apresentavam os jogadores que se deseja identificar. A quantidade de cada objeto que foi usada para o treinamento está representada na Tabela 2. A rede neural processou ate obter um resultado considerado satisfatório para a aplicação, sendo aproximadamente 40 mil etapas de execução do treinamento.

Tabela 2 – Quantidade objetos usados para treinamento

Objeto	Quantidade
Jogador	1422
Bola	163
Traves	136

Figura 3 – Imagens dos jogadores



Os objetos que desejava identificar eram a bola, as traves das goleiras, e os jogadores. Num primeiro instante tentou-se usar as redes neurais já para classificar a que time cada jogador pertencia, porém, aplicada desta forma ela apresentou muitos falsos positivos, confundindo os jogadores entre as equipes, o que fez com que houvesse a necessidade de fazer essa classificação externamente à rede. A técnica utilizada para esse fim está presente na seção 4.3.

Esses modelos de redes neurais fazem o reconhecimento de objetos e os destacam através de retângulos, delimitados por dois pontos, um superior esquerdo e um inferior direito. No entanto, cada imagem é processada de maneira individual, ou seja, para a utilização nos vídeos da partida, se aplicava a identificação de objetos feito pela rede neural para cada *frame*, havendo a necessidade de um rastreamento dos mesmos objetos que seriam identificados nos *frames* seguintes. Os métodos aplicados para rastreamento estão presentes na seção 4.4.

4.3 Metodologia para classificação dos jogadores Entre os times

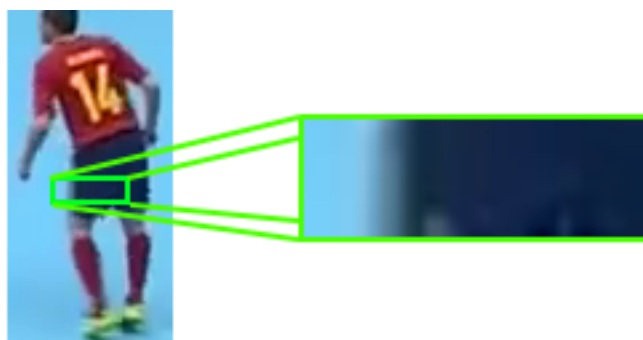
Como visto os jogadores foram identificados independentemente dos times, ou seja, todos eram reconhecidos mas não se sabia os times em que estavam defendendo. Então houve a necessidade de um classificador para esse jogadores, e o método usado foi baseado no algoritmo *Kmeans*, que *clusteriza* as cores pertencentes à área da imagem onde o jogador foi identificado. Essa área ainda foi reduzida para se utilizar apenas o que seria o tronco do jogador, ou seja, a parte definida pela camiseta dele. No entanto, essa região ainda apresentou muito ruído como pode ser visto na Figura 4, por exemplo as cores, da quadra, do número da camiseta e do calção, poderiam afetar na clusterização. Para obter melhores resultados então na hora de classificar os times, a região foi mudada para a área onde o calção do jogador está definido, como pode ser visto na Figura 5. Essa área apresentou uma quantidade menor de ruídos e foi menos afetada pelas inclinações

dos corpos dos jogadores.

Figura 4 – Região da camiseta jogador



Figura 5 – Região da calção jogador



Como visto na subseção 2.1.2, o *Kmeans* é um algoritmo de clusterização que define um número de centroides correspondentes à quantidade de *clusters* que se deseja identificar. Assim, para as imagens que foram analisadas, utilizamos dois centroides, fazendo com que a cor predominante sempre fosse identificada e não houvesse interferência devido ao diferente tipo de posicionamento do jogador. Algumas interferências conhecidas são por exemplo sobreposição do braço sobre a calção ou quando a região de interesse pega uma parte da quadra.

Para evitar a interferência externa na hora de determinar a cor dos uniforme, foi utilizado então uma variação do algoritmo BSAS, cujo objetivo era pegar os resultados obtidos pelo *Kmeans*, e *clusterizá-los* determinando um valor limite para a distância de cada centroide, ajustando sempre que um novo individuo fosse inserindo no grupo. Esse método foi usado para que fosse possível reconhecer a frequência de cada cor. As duas cores mais frequentes então, determinam a cor do uniforme de cada um dos times. Essa frequência foi retirada nos 20 primeiros *frames* de cada partida, onde os 2 times estão sempre presentes, poupando processamento nos *frames* seguintes e evitando a mudança na frequência das cores. Quando um jogador novo surgia, caso nenhuma das duas cores

mais frequentes fossem reconhecidas, se mantinha o jogador com o time indeterminado, até que em suas detecção seguintes, fosse possível determinar a cor do time ao qual ele pertencia.

4.4 Metodologia para rastreamento e suavização dos jogadores entre os frames

Como visto anteriormente na seção 4.2, a identificação de objetos faz o reconhecimento, porém não é capaz de definir se o objeto é o mesmo representado nas imagens em sequência. Para solucionar isso se aplicou o rastreamento dos objetos ao longo dos *frames* externo à rede. O rastreamento então consistiu primeiramente em verificar a menor diferença na distância entre os pontos de um objeto identificado em um *frame* em relação ao seu anterior. Os resultados obtidos apenas por essa distância já foram suficiente para se manter o rastreamento dos jogadores quando estão isolados. Porém quando dois jogadores, em um *frame* anterior, estão próximos a mesma detecção no *frame* seguinte, poderia acontecer uma troca no rastreamento, visto que ambos são rastreados para uma mesma caixa da nova detecção. Para solucionar essa troca a somatória das distâncias entre jogadores, já pre detectados em *frames* anteriores, e as identificações dos objetos em um novo *frame* devia ser a menor possível, sendo assim nunca dois jogadores seriam rastreados para a mesma caixa, este método então garantiu o rastreamento dos jogadores quando todos estivessem presentes na imagem.

Quando um objeto não é reconhecido em um *frame* subsequente a outros em que o mesmo foi identificado, usa-se uma técnica de predição com o objetivo de cobrir essa falha. A técnica consiste em determinar uma velocidade para cada objeto. Para a bola, que é um objeto que não possui uma forma de alterar sua velocidade sem uma intervenção externa, essa técnica consiste em calcular diferença entre os pontos inferiores de cada uma dos *frames* onde ela foi reconhecida, definindo assim um vetor velocidade para ela. Ou seja, nesse cálculo são obtidos dois resultados em relação à subtração desses pontos - um é o deslocamento no eixo x e outro no eixo y . O calculo esta representado a seguir:

$$vel_x = x_i - x_{i-1}$$

$$vel_y = y_i - y_{i-1}$$

Para determinar a velocidade dos jogadores, foi necessário obter uma média da velocidade em que ele se movimenta durante os *frames*, visto que grandes variações em sua caixa de detecção poderia ocorrer por uma interferência, que fizesse com que a velocidade em relação a dois *frames* mudasse de maneira brusca. As interferências poderiam ser, por exemplo, o esticamento da caixa de um jogador, que poderia ocorrer por alguma inclinação em seu tronco, ou quando duas caixas se aglutinam, o que faria com que houvesse uma variação na velocidade que não representasse o real movimento do jogador. Essa média

foi determinada pelos 10 últimos *frames* em que o jogador foi detectado e o cálculo pode ser visto a seguir:

$$vel_y = \sum_{i=n}^{n-10} x_i - x_{i-1}$$

Como pode ser visto, não se determinou uma velocidade em relação ao eixo y para os jogadores, pois a variação nesse eixo fez com que a utilização do método de predição em relação a ele não fosse uma boa aplicação. Porém, para evitar mudanças bruscas ocasionadas por interferências, se manteve um valor para a altura de cada jogador, fazendo com que a redução da caixa de um jogador não fosse aplicada de imediato, e sim de maneira gradual, consequência a isso o tamanho de um jogador se mantém mais próximo do real. A altura então foi calculada através de uma média das alturas dos últimos 10 *frames*.

$$vel_y = \sum_{i=n}^{n-10} y_{2,i} - y_{1,i}$$

Tanto a técnica de predição quanto determinar a altura do jogador foram importantes para corrigir interferências que gerariam um resultado equivocado na identificação de objetos. Por exemplo, na Figura 6, está representada a identificação de objetos sem a correção da altura e com a correção, o que faz com que a caixa do jogador tivesse um valor mais próximo do que se esperava.

Figura 6 – Exemplo da correção altura

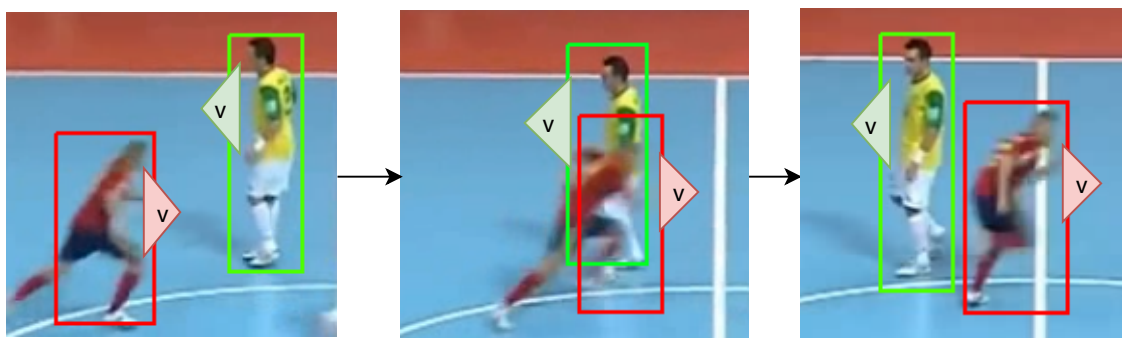


Uma correção feita pela predição pode ser representada na ocasião onde duas caixas se aglutinam. Um exemplo dessa aglutinação pode ser visto na Figura 7. As caixas dos jogadores se juntam no *frame* do meio, detectando apenas um jogador, e após o jogador de vermelho voltar a ser detectado sendo necessário um recálculo para o time dele. Na Figura 8 é possível ver como a correção através da velocidade atua, onde é previsto a posição do jogador no *frame* do meio e em sequência ele volta a ser identificado com o mesmo time ao qual pertencia.

Figura 7 – Exemplo sem correção da velocidade



Figura 8 – Exemplo com correção da velocidade



4.5 Metodologia para detecção de eventos

Para a detecção de eventos, primeiramente foram selecionados quais poderiam ser detectados com os resultados obtidos pela identificação de objetos. A partir dessa análise chegou-se à conclusão que poderiam ser detectados os seguintes eventos:

- Posse de bola entre as equipes.
- Passes certos e errados de cada equipe.
- Interceptações feitas por equipes.
- Lances de disputa de bola.
- Desarmes.
- Jogadas Individuais.
- Chutes a gol.

Chute a gol é um lance incluso que pode ser identificado pois o algoritmo é capaz de reconhecer todos os objetos necessários para fazer o reconhecimento. Porém a detecção

desse evento não foi implementada por falta de tempo e pelo fato do foco do trabalho ser relacionado em lances jogador x jogador.

Como pode ser visto, o levantamento de estatísticas para cada jogador individualmente não pode ser feito, pela falta de um rastreamento integral para cada um deles, visto que quando o jogador sai do ângulo de visão da câmera, não é possível determinar aonde ele está. Apesar de que, de maneira geral, a detecção de eventos aqui proposta poderia ser aplicada em uma ferramenta que mantivesse o rastreamento individual de todos jogadores, sendo assim possível a detecção individual através do que foi aqui proposto.

A detecção de eventos então consistiu em definir uma máquina de estados para a bola, através do posicionamento dela e dos jogadores na imagem. Esse modelo foi assim definido porque a bola é o elemento determinante da grande maioria dos eventos. Para isso foram definidos três estados - enquanto a bola está em jogo ela fica alterando entre os estados, e um evento é caracterizado por uma sequência desses estados e os atributos pertencentes aos indivíduos que determinam esse estado. A seguir são listados cada um dos estados e o que os define, e na Figura 9 é possível ver uma representação da máquina.

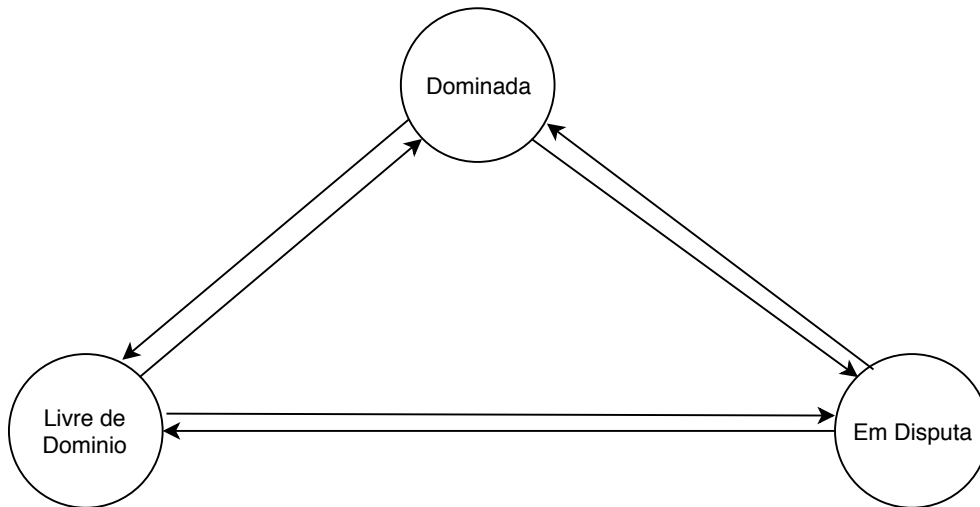
- Dominada: Quando a bola está próxima a um jogador por um determinado tempo, esse jogador se encontra com o domínio dela. O estado de domínio foi usado também como técnica de rastreamento da bola, definindo que um jogador que a dominasse durante uma sequência de *frames*, mantivesse ela junto a caixa de detecção dele, durante os *frames* que ela não fosse identificada.
- Em disputa: Quando a bola está próxima de dois jogadores por um determinado tempo, a bola está em disputa entre eles.
- Livre de Domínio: Quando a referência da bola na imagem está distante de qualquer jogador.

Nas seções que seguem serão apresentados cada um dos eventos detectados e a sequência de estados que os definem, assim como uma breve descrição justificando o motivo dele ser definido por aquela sequência.

4.5.1 Posse de bola entre as equipes

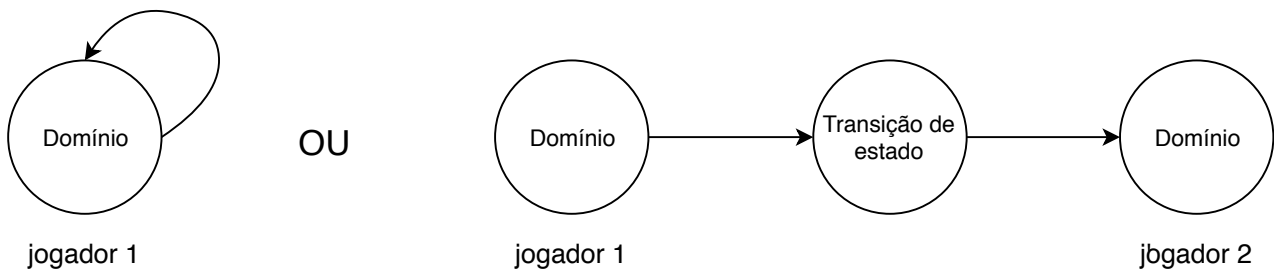
Uma posse de bola é determinada por uma sequência de domínios, feita por apenas um jogador. A partir da primeira sequência, a cada novo domínio subsequente desse jogador, a posse de bola do time ao qual ele pertence é somada em 1 unidade. Considerando que a bola pode tomar uma distância do jogador após um toque, ou outro jogador vir a disputar o lance, ocorreria uma transição de estado. Caso após o estado transicional retornasse ao estado de domínio pelo mesmo jogador, esse meio tempo, onde a bola se encontra em disputa ou livre, é considerado como domínio do jogador. Na Figura 10 estão

Figura 9 – Estados pertencentes à bola



representadas as duas possibilidades de sequência de estados que determinam a posse de bola para um jogador.

Figura 10 – Sequência de estados que determinam posse de bola

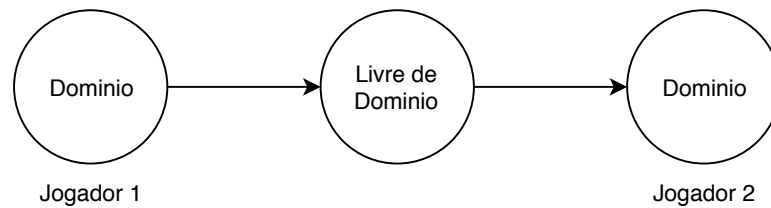


4.5.2 Passes certos, errados e intercepções de cada equipe

Um passe foi reconhecido com a troca de domínio entre jogadores, caso o passe seja certo, o time dos jogadores que dominaram a bola nos diferentes momentos seria igual, caso o time fosse diferente, o passe seria considerado errado. A intercepção foi determinada pela mesma sequência de um passe errado, porém ela influencia no time que recuperou a bola. A sequência de de estados para determinar um passe é variada, no primeiro caso a bola esta sob domínio de um jogador, fica livre de domínio, e volta a ser dominada por outro jogador.

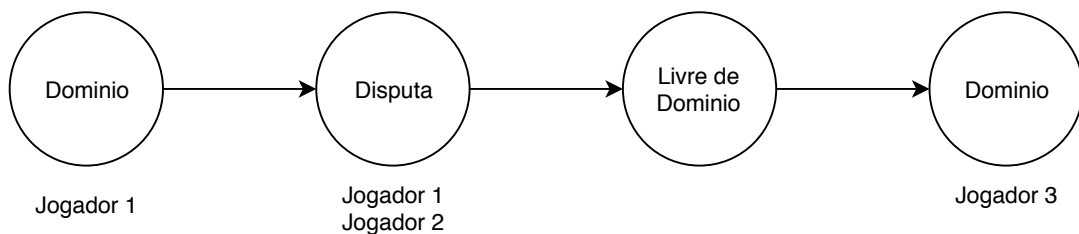
No segundo caso a bola esta dominada por um jogador, e passa a ser disputada com outro, se houver uma sequência onde a bola fica livre de domínio, e só então volta a ser dominada por um novo jogador, diferente dos que estavam disputando a bola, é consi-

Figura 11 – Sequência de estados que determinam passes 1



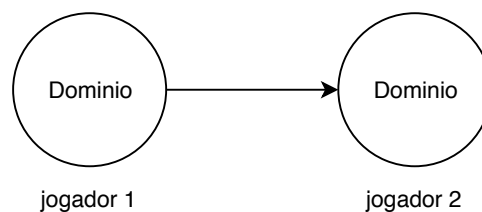
derado um passe, o passe é considerado certo pelo time do jogador que recebeu/dominou por ultimo a bola. A seguir um exemplo desta sequência.

Figura 12 – Sequência de estados que determinam passes 2



A ultima sequência de estados que pode definir um passe é a troca de domínio de imediato, onde os jogadores pertencem ao mesmo time. Isso pode ocorrer pela falta da detecção da bola em alguns *frames* ou pelos jogadores estarem muito próximo na hora do passe. Nesse caso, caso o jogador ao final da sequência seja de outro time não é considerado um passe certo e sim, uma roubada de bola definido na subseção 4.5.3.

Figura 13 – Sequência de estados que determinam passes 3

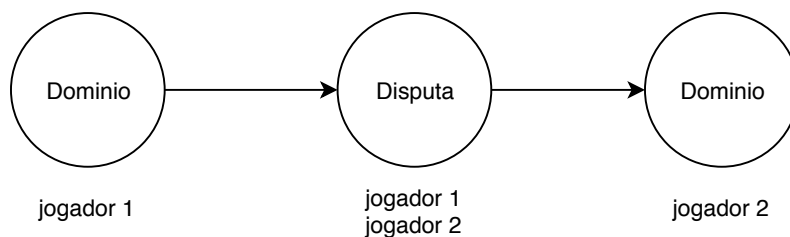


Outra forma de detectar um passe errado foi verificar se ao final de um lance a bola esta livre de domínio, isso é justificado pois se considera que esse tenha sido um passe para fora, sendo assim um passe errado feito pelo jogador que dominou por ultima a bola.

4.5.3 Desarmes e Jogadas individuais

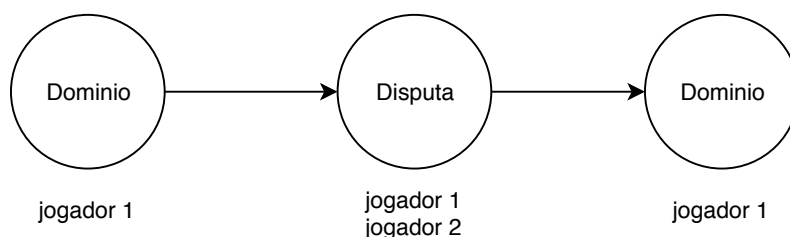
Um desarme ocorre quando um jogador perde a bola para outro jogador do time adversário. A sequência de estados que determina um desarme, tem início com um domínio pertencente a um jogador, e ao final da sequência o domínio pertence a um jogador do time adversário. Existem três sequências possíveis para a ocorrência de um desarme, uma delas foi representada na Figura 12, onde após a disputa a bola é dominada por um jogador com time diferente do jogador domínio inicial, caracterizando então a roubada de bola. A Figura 13 representa outra possibilidade de roubo de bola, onde caso o jogador 2 venha ser de time diferente do jogador 1 significa que aconteceu um desarme. A última sequência de estados que pode definir um desarme esta representada na Figura 14, onde um jogador possui o domínio da bola entra em uma disputa por ela, e ao final o domínio pertence a outro jogador de time diferente.

Figura 14 – Sequência de estados que determinam um desarme



Ao contrario do desarme, uma jogada individual ocorre quando um jogador esta sobre o domínio da bola, entra em uma disputa por ela, e ao final consegue sair com o domínio ou conseguindo a execução de um passe para seu companheiro. A ocorrência de uma jogada individual então pode ser representada pela Figura 12, que sobre as mesmas condições de quando essa sequência é considerada um passe certo, também é considerado uma jogada individual pelo jogador da disputa. Outra sequência de estados que determina uma jogada individual pode ser visto na Figura 15.

Figura 15 – Sequência de estados que determinam uma jogada individual



4.5.4 Disputa de Bola

A fim de separar lances que podem ter ocorrido disputas de bolas mais fortes, foi utilizado informações da identificação de objetos, e não estados da bola como os outros eventos. Quando uma disputa é mais forte, onde a probabilidade de acontecer uma falta é maior, as caixas de detecção dos jogadores tendem a se juntar e se tornar apenas uma. Levando isso em consideração, o evento de disputa pode ser detectado com a aglutinação das caixas dos jogadores próximo a bola, sendo assim um deles deixa de ser detectado. Através de uma detecção de colisão entre as caixas em *frames* anteriores é possível definir essa junção e determinar um lance mais forte entre os jogadores. Na Figura 16 é possível ver uma dessas disputas, o jogador de branco representado pela caixa de verde não foi identificado nesse *frame*, a partir da técnica de predição e altura ele foi reconhecido ali. Como o jogador não foi identificado e uma colisão de retângulos esse lance foi caracterizado como passível de falta.

Figura 16 – Lance passível de falta



4.6 Ferramentas para implementação da aplicação

Para a implementação da aplicação, foi utilizado a linguagem python, por oferecer ferramentas que favoreceram o desenvolvimento da proposta. Para identificação de objetos se usou a biblioteca *TensorFlow*, por facilitar a aplicação de redes neurais e fornecer modelos pré treinados para esse fim. A biblioteca *sklearn* foi usada para a aplicação do algoritmo *k-means*, aplicado na identificação de cores nos uniformes dos jogadores. Para o processamento de imagem, foi utilizado a biblioteca *OpenCV* por ser uma ferramenta desenvolvida para esse fim e bastante usada na área garantindo uma boa quantidade de conteúdo online, que auxiliou o uso dela.

4.7 Avaliação dos resultados

Para testar a aplicação foi escolhido três vídeos de partidas de futebol de salão que aconteceram na copa do mundo de futsal de 2012. Na Tabela 3 esta identificada cada uma das partidas escolhidas. A aplicação foi usada nos vídeos, num primeiro momento foi verificado os objetos reconhecidos pela identificação de objetos. Foi testado também o rastreamento verificando onde ele não foi capaz de acompanhar os jogadores e a bola. E, por fim, uma avaliação na classificação verificando precisão dela.

Tabela 3 – Partidas utilizadas para avaliação

Identificador	Partida
Jogo 1	Brasil x Espanha
Jogo 2	Brasil x Japão
Jogo 3	Espanha x Irã

Para a detecção de eventos, primeiramente foi feito a correções sobre a identificação de objetos para que o erro não fosse propagado. Então foi feita uma avaliação sobre a definição dos estados da bola. E por fim foi verificado cada um dos lances dos vídeos junto à a aplicação da detecção de eventos, verificando se os lances foram identificados, e se foram identificados de maneira correta.

5 RESULTADOS OBTIDOS

Nesta seção serão apresentados os resultados de cada uma das áreas exploradas pelo trabalho. Na área de identificação de objetos serão vistos resultados obtidos pelo reconhecimento de objetos junto ao rastreamento, e resultados obtidos pela classificação em time dos jogadores. Na detecção de eventos os resultados obtidos foram em relação à definição dos estados na máquina de estados, e os eventos identificados.

5.1 Identificação de objetos

Para testar a identificação de objetos nos dois diferentes modelos de rede neural convolucional, *Faster R-CNN* e *SSD*, as duas foram aplicadas sobre uma sequência de 200 *frames*. Nestes *frames*, um total de 1250 objetos deveriam ser reconhecidos como jogadores. A aplicação do modelo *Faster R-CNN* teve um reconhecimento dos objetos em mais frames, apresentando uma grande diferença em relação ao modelo *SSD*. Apesar do modelo *SSD* ser mais rápido do que o *Faster R-CNN*, seria inviável cobrir a identificação de objetos em sequências muito grandes de *frames*, o que levou a ser escolhido o modelo *Faster R-CNN* para dar continuidade ao trabalho. Na Tabela 4 é possível ver os resultados obtidos.

Tabela 4 – Resultados redes neurais convolucionais

Modelo	Precisão (%)
<i>Single Shot Multibox Detector (SSD)</i>	38%
<i>Faster Region - Convolutional Neural Network (Faster R-CNN)</i>	97%

Estes resultados foram obtidos em relação à detecção de jogadores, além deles, ainda se desejava reconhecer a bola e as traves das goleiras. Os resultados para esses objetos, obtidos pelo modelo *Faster R-CNN*, estão representados na Tabela 5 junto aos jogadores. Para esses objetos, foram usados menos exemplos para o treinamento por sua menor frequência nas imagens, o que pode explicar os resultados inferiores. A quantidade de imagens usados para treinamento de cada objeto pode ser visto Tabela 2. É possível concluir ainda que a detecção da bola, por ser um objeto pequeno e com poucas características, produziu um resultado bem inferior em relação aos demais.

Na Figura 17 é possível observar o grau de confiança que a ferramenta *tensorboard*, que acompanha a biblioteca *TensorFlow*, fornece durante o treinamento dos modelos de redes neurais. Nela, é possível observar o resultado após 38 mil etapas aplicadas pela rede neural. É possível notar que o grau de confiança para identificação dos objetos na imagem aplicada chegou muito próximo de 100%.

Tabela 5 – Resultados modelo *Faster R-CNN*

Objeto	Precisão (%)
Jogadores	97%
Bola	47%
Traves	91%

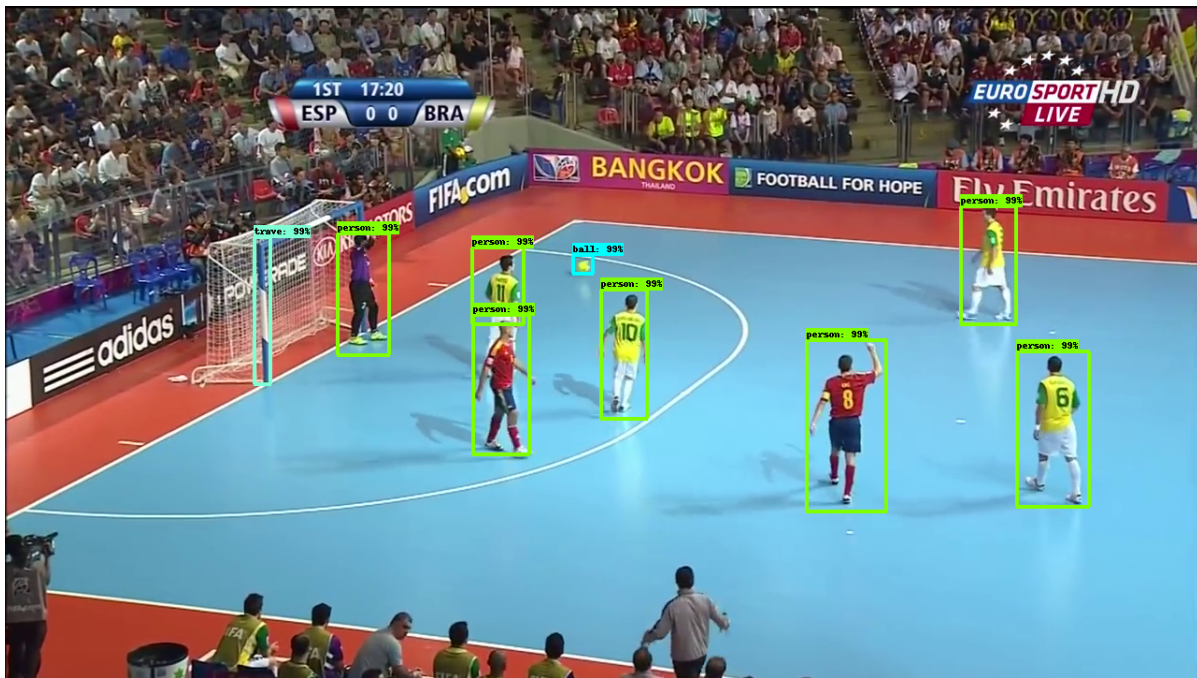


Figura 17 – Nivel de confiabilidade fornecido pelo TensorFlow

5.2 Classificação dos jogadores em times

A classificação foi testada nas 3 partidas, sendo que a quantidade de testes está representada na Tabela 6, assim como uma porcentagem representando a precisão que a classificação teve.

Em geral, os erros apresentados em todos os jogos ocorreram sobre a deformação da caixa do jogador que deveria ser classificado. Deformações normalmente causadas por efeitos de obstruções, geradas ou pelo limite da imagem ou por outro jogador. No jogo 2 ainda é possível ver uma queda na porcentagem de acertos, o que é justificável pois na partida o uniforme de um dos times era muito semelhante à cor da quadra. Isso fez com que, em algumas classificações, a região de interesse para definir a cor do time do jogador considerasse cores da quadra, classificando-o de forma equivocada.

Tabela 6 – Resultados classificação entre times

Jogo	Total de classificações	Corretos	Precisão
Jogo 1	1179	1110	94%
Jogo 2	1186	1084	91%
Jogo 3	1154	1096	95%

5.3 Rastreamento dos jogadores

O rastreamento é a forma que a aplicação utiliza para acompanhar os objetos ao longo dos *frames* no qual eles estão presentes, além de corrigir em sequências onde ele não foi identificado. Para avaliar a técnica implementada, foi verificado quantas vezes os jogadores foram perdidos e não foram encontrados em menos de 10 *frames*. Para evitar que jogadores que tenham saído da imagem entrem nessa contagem, foram considerados apenas jogadores que, na ultima identificação feita, estavam a uma certa distância das bordas laterais. Esse modelo de avaliação foi aplicado sobre cerca de 1000 jogadores em cada partida, e os resultados podem ser vistos na Tabela 7.

Tabela 7 – Resultados rastreamento jogadores

Jogo	Total de classificações	Corretos	Jogadores Perdidos %
Jogo 1	1194	954	80%
Jogo 2	1131	919	81%
Jogo 3	1168	972	83%

O maior problema do rastreamento foi em relação à câmera que, por não ser fixa, em alguns momentos com o movimento brusco no acompanhando da bola, acarretou que todos jogadores fossem perdidos. Mesmo sob essas condições os jogadores foram perdidos apenas 19% das vezes aproximadamente, o que garantiu que o rastreamento utilizado não fosse considerado ruim.

5.4 Definição dos estados e domínio

Os resultados obtidos pela definição dos estados estão fortemente ligados à identificação da bola feita pela identificação de objetos e o rastreamento dela, visto que ela é o elemento fundamental para a definição. No geral, quando a bola está em livre domínio, a identificação apresenta os melhores resultados já que é provável que ela esteja visível

na imagem. Sob domínio ou disputa a visão dela, pela câmera, é muitas vezes encoberta pelo jogador que a está conduzindo.

Os resultados obtidos para a definição de quem está no domínio da bola variam conforme a sequência de estados que ocorre. Quando o domínio é livre de uma disputa, os resultados apresentam-se bons, atingindo valores acima de 90%. Quando a sequência de estados envolve uma disputa, há uma queda nesses valores devido à obstrução da bola, o que faz com que o rastreamento dela através dos domínios seja trocado entre os jogadores. Na Tabela 8 é possível a diferença na precisão em relação à quem domina a bola sem a ocorrência de uma disputa e com a ocorrência de uma disputa.

Tabela 8 – Precisão definição do domínio

Forma de domínio	Precisão
Domínio sem envolvimento de uma disputa	94%
Domínio envolvendo uma disputa	80%

Os principais motivos que causaram erro na definição do jogador que está dominando a bola foram a falta da identificação da bola, seja por obstrução ou por falha no reconhecimento, ou erro ocasionado pela perspectiva da câmera, que fez parecer, em alguns momentos, que a bola estava dominada por um jogador distante dela.

5.5 Detecção de eventos

A detecção de uma posse de bola esta fortemente ligada a definição do estado de domínio, ou seja a precisão deste evento pode ser representado pela Tabela 8. Garantindo então que o resultados em relação a esse evento sejam considerados suficientemente bons. Para a detecção dos outros eventos propostos neste trabalho, foram usados um total de 1629 eventos, retirados das três partidas nas quais a aplicação foi executada. A Tabela 10 apresenta o número de vezes em que cada evento apareceu nos testes, o numero de vezes em que cada um foi detectado, o numero de vezes que foram detectados de maneira errada e a quantidade de vezes em que os eventos não foram detectados. A diferença nas quantidades de testes para cada evento, é devida à frequência dos eventos nos vídeos, por isso alguns deles foram testados menos vezes que outros.

Alguns eventos apresentaram uma grande quantidade de erros chegando próximo a 30% dos lances detectados como errados. Considerando que a maior parte dos erros são propagados pela definição de quem esta com o domínio da bola, melhorando essa definição, potencialmente diminuiria os erros apresentados pela detecção de eventos.

A Tabela 10 representa a matriz de confusão da detecção dos eventos. Nesta tabela passes errados e interceptações foram colocados juntos, pois, onde ocorreram os erros

Tabela 9 – Resultados obtidos pela detecção de eventos

Evento	Total em video	Detectados	Detectados Errado	Não detectados
Passes Certos (PC)	1024	953	60	71
Passes Errados (PE)	164	146	28	18
Interceptações (INT)	109	92	28	17
Roubo de Bola (RB)	101	85	27	16
Jogadas Individuais (JI)	231	207	48	24

detectados desta maneira, eles estavam relacionados. Como pode ser visto, um número significativo de passes certos foram considerados passes errados. Uma justificativa para esse resultado está no erro da definição do domínio pois, em alguns momentos, quando a bola passa próxima de um jogador do time adversário ela foi considerada como sob domínio dele, caracterizando assim um passe errado. A mesma ideia se aplica a passes certos que foram considerados "roubadas de bola". Observa-se também que muitos passes errados foram considerados "roubadas de bola", o que aconteceu em momentos em que a bola não era identificada até aparecer sob domínio de outro jogador. Lances individuais e roubadas de bolas foram confundidos quando a bola fica obstruída ao final do lance, fazendo com o que o domínio dela fosse representado pelo jogador que perdeu a disputa.

Tabela 10 – Matriz de confusão detecção de evento

		Real			
		PC	PE/INT	RB	JI
Detectado	Passes Certos (PC)	-	0	0	0
	Passes Errados e Interceptação (PE/INT)	13	-	7	0
	Roubo de Bola (RB)	7	10	-	9
	Jogadas Individuais (JI)	3	0	8	-

5.6 Detecção de lances possíveis de faltas

Como visto na subseção 4.5.4, foi proposta uma maneira de detectar lances que podem ter gerado faltas. Os resultados obtidos por essa técnica podem ser visto na Tabela 11, dividindo esses lances em lances que realmente aconteceram faltas, lances normais sem falta, e lances que merecem uma melhor avaliação se ocorreu ou não uma falta.

Tabela 11 – Resultados obtidos pela detecção de eventos

Tipo de lance	Detecções
Lance normal	12
Lance passível de avaliação	48
Lance com ocorrência de falta	13

Num total de 15 ocorrências de faltas marcadas nos vídeos, 13 delas foram detectadas pela aplicação dos métodos apresentados. Eles não garantem que tenha ocorrido uma falta, mas separam lances onde a disputa foi mais forte entre os jogadores foi mais acirrada, e a possibilidade da ocorrência de um contato entre eles, o que geraria uma falta, é maior.

6 CONCLUSÕES

Num primeiro instante, o objetivo do trabalho era fazer a gravação das partidas através do posicionamento de câmeras que garantissem um melhor resultados para as técnicas aplicadas. Isso não foi possível devido à indisponibilidade de equipamentos necessários para as gravações. Por consequência, os vídeos utilizados não foram os ideais para o que se desejava implementar porém, ainda assim, foi possível a avaliação de técnicas com potencial de utilização para esta finalidade.

Dessa maneira, o trabalho com os vídeos que foram obtidos consistiu em fazer um levantamento de técnicas e propor novos métodos para a identificação de objetos e detecção de eventos em partidas de futebol e futebol de salão. Os resultados obtidos pela identificação de objetos foram considerados adequados para a aplicação em uma ferramenta, sendo feito um conjunto de testes que garantem isso. O mesmo pode ser dito para o rastreamento dos jogadores e a classificação em times, através do método proposto. A detecção de eventos propõe uma nova maneira para fazer o reconhecimento de eventos e, apesar da baixa quantidade de testes aplicados, os resultados sugerem que a proposta tem potencial para ser usada em uma ferramenta de aplicação prática em jogos reais.

Em trabalhos futuros estas mesmas técnicas poderão ser aplicadas, porém com a utilização de vídeos com formato e qualidade que favoreçam os resultados da aplicação, finalmente, com os resultados obtidos com a aplicação desses

vídeos, poderemos concluir se as técnicas desenvolvidas são viáveis de utilização em uma ferramenta que auxilie o levantamento de dados de uma partidas de futebol e futebol de salão, ajudando times, jogadores e árbitros na obtenção de um melhor desempenho.

REFERÊNCIAS

- FREUND, Y.; SCHAPIRE, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. **Journal of computer and system sciences**, Elsevier, v. 55, n. 1, p. 119–139, 1997. Citado na página 24.
- GIRSHICK, R. et al. Rich feature hierarchies for accurate object detection and semantic segmentation. In: IEEE COMPUTER SOCIETY. **Proceedings of the IEEE conference on computer vision and pattern recognition**. Washington, 2014. p. 580–587. Citado na página 24.
- GOLDSCHMIDT, R. R. Uma introdução à inteligência computacional: fundamentos, ferramentas e aplicações. **Rio de Janeiro Brasil: IST-Rio**, p. 32, 2010. Citado 2 vezes nas páginas 21 e 22.
- JAHNE, B. **Computer vision and applications: a guide for students and practitioners**. [S.l.]: Academic Press, 2000. Citado na página 22.
- KE, Y.; SUKTHANKAR, R.; HEBERT, M. Event detection in crowded videos. In: IEEE. **Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on**. Rio de Janeiro, 2007. p. 1–8. Citado na página 23.
- KHAN, A. et al. **Soccer Event Detection**. 2018. 119-129 p. Citado 3 vezes nas páginas 19, 23 e 27.
- LECUN, Y. et al. Lenet-5, convolutional neural networks. **URL: <http://yann.lecun.com/exdb/lenet>**, p. 20, 2015. Citado na página 24.
- LIU, J. et al. Automatic player detection, labeling and tracking in broadcast soccer video. **Pattern Recognition Letters**, Elsevier, v. 30, n. 2, p. 103–113, 2009. Citado na página 27.
- LIU, W. et al. Ssd: Single shot multibox detector. In: SPRINGER. **European conference on computer vision**. Amsterdam, 2016. p. 21–37. Citado 2 vezes nas páginas 23 e 25.
- LLOYD, S. Least squares quantization in pcm. **IEEE transactions on information theory**, IEEE, v. 28, n. 2, p. 129–137, 1982. Citado na página 22.
- MATSUGU, M. et al. Subject independent facial expression recognition with robust face detection using a convolutional neural network. **Neural Networks**, Elsevier, v. 16, n. 5-6, p. 555–559, 2003. Citado na página 24.
- MCCULLOCH WARREN S E PITTS, W. Cálculo lógico das idéias imanentes à atividade nervosa. **O boletim de biofísica matemática**, Springer, v. 5, n. 4, p. 115–133, 1943. Citado na página 21.
- MOESLUND, T. B.; THOMAS, G.; HILTON, A. **Computer vision in sports**. [S.l.]: Springer, 2014. Citado na página 22.
- MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre aprendizado de máquina. **Sistemas inteligentes-Fundamentos e aplicações**, v. 1, n. 1, p. 32, 2003. Citado na página 21.

- MOREIRA, G. C. G. **Reconhecedor de Objetos em Vídeos Digitais para Aplicações Interativas**. Tese (Doutorado) — PUC-Rio, 2008. Citado 2 vezes nas páginas 23 e 24.
- PARKER, J. R. **Algorithms for image processing and computer vision**. [S.l.]: John Wiley & Sons, 2010. Citado na página 22.
- REN, S. et al. Faster r-cnn: Towards real-time object detection with region proposal networks. In: **Advances in neural information processing systems**. Montreal: [s.n.], 2015. p. 91–99. Citado na página 25.
- SHALEV-SHWARTZ, S.; BEN-DAVID, S. **Understanding machine learning: From theory to algorithms**. [S.l.]: Cambridge university press, 2014. Citado na página 22.
- SOO, S. Object detection using haar-cascade classifier. **Institute of Computer Science, University of Tartu**, 2014. Citado na página 23.
- SPAGNOLO, P. et al. An unsupervised approach for segmentation and clustering of soccer players. In: IEEE. **Machine Vision and Image Processing Conference, 2007. IMVIP 2007. International**. Kildare, 2007. p. 133–142. Citado na página 29.
- SZELISKI, R. **Computer vision: algorithms and applications**. [S.l.]: Springer Science & Business Media, 2010. Citado na página 22.
- TAVASSOLIPOUR, M.; KARIMIAN, M.; KASAEI, S. Event detection and summarization in soccer videos using bayesian network and copula. **IEEE Transactions on circuits and systems for video technology, IEEE**, v. 24, n. 2, p. 291–304, 2014. Citado na página 29.
- TOVINKERE, V.; QIAN, R. J. Detecting semantic events in soccer games: Towards a complete solution. In: IEEE. **Multimedia and Expo, 2001. ICME 2001. IEEE International Conference on**. Tokyo, 2001. p. 833–836. Citado na página 29.
- VIOLA, P.; JONES, M. Rapid object detection using a boosted cascade of simple features. In: IEEE. **Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on**. Kauai, 2001. v. 1, p. I–I. Citado 3 vezes nas páginas 23, 24 e 33.
- YU, X. et al. Trajectory-based ball detection and tracking with applications to semantic analysis of broadcast soccer video. In: ACM. **Proceedings of the eleventh ACM international conference on Multimedia**. Berkeley, 2003. p. 11–20. Citado 2 vezes nas páginas 23 e 28.

ÍNDICE

BSAS, 29, 32, 35

COCO, 33

FIFA, 19

MCMC, 28, 32

Faster R-CNN, 25, 33, 45, 46

SSD, 25, 27, 31, 33, 45

TCC1, 31

TCC2, 31