

UNIVERSIDADE FEDERAL DO PAMPA

Sabrina Carlé Winckler

**Desenvolvimento de um Sistema Inteligente  
para Segurança Rural com Arquitetura  
Flexível**

Alegrete  
2021



Sabrina Carlé Winckler

## **Desenvolvimento de um Sistema Inteligente para Segurança Rural com Arquitetura Flexível**

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

Orientador: Prof. Dr. Marcelo Resende Thielo

Alegrete  
2021



**Sabrina Carlé Winckler**

Desenvolvimento de um sistema inteligente para segurança rural com arquitetura flexível

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Software da Universidade Federal do Pampa, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Trabalho de Conclusão de Curso defendido e aprovado em: 14 de outubro de 2021.

Banca examinadora:

---

Prof. Dr. Marcelo Resende Thielo

Orientador

UniPampa

---

Prof.<sup>a</sup> Dra. Aline Vieira de Mello

UniPampa

---

Prof. Me. Eliezer Soares Flores

UniPampa

---



Assinado eletronicamente por **MARCELO RESENDE THIELO, PROFESSOR DO MAGISTERIO SUPERIOR**, em 14/10/2021, às 19:12, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

---



Assinado eletronicamente por **ALINE VIEIRA DE MELLO, PROFESSOR DO MAGISTERIO SUPERIOR**, em 14/10/2021, às 20:17, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

---



Assinado eletronicamente por **ELIEZER SOARES FLORES, PROFESSOR DO MAGISTERIO SUPERIOR**, em 21/10/2021, às 15:16, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

---



A autenticidade deste documento pode ser conferida no site [https://sei.unipampa.edu.br/sei/controlador\\_externo.php?acao=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](https://sei.unipampa.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **0625192** e o código CRC **CD5E7F69**.

---

## RESUMO

O reconhecimento de eventos suspeitos em vídeos é um dos maiores desafios da área de visão computacional. O presente trabalho aborda a identificação de situações anormais em um cenário pouco estudado até então, a zona rural. Este cenário foi selecionado devido à crescente taxa de criminalidade nestas zonas e os perigos à sociedade que crimes como estes desencadeiam. Apesar de grande parte dos métodos propostos para reconhecimento de eventos anormais se basearem no aprendizado supervisionado, existem algumas dificuldades na utilização deste tipo de aprendizado na tarefa de vigilância. Sendo assim, propomos a utilização de um filtro de eventos que receba dados da *clusterização* de trajetórias a partir de um algoritmo genético combinado com o sistema supervisionado de rastreamento e detecção de objetos *Yolo* para o reconhecimento de situações anormais em vídeos. Um *dataset* para o cenário escolhido foi construído utilizando vídeos da internet que se mostraram bons candidatos para o estudo. Foram geradas simulações de trajetórias artificiais a fim de comparar os resultados com os dados reais obtidos. Os resultados deste estudo se mostraram promissores pois, apesar do custo computacional do algoritmo genético e da sensibilidade do rastreador, o filtro conseguiu delinear os eventos anormais de forma satisfatória tanto com os dados artificiais quanto com os reais. Atualmente há um grande potencial em impulsionar os resultados desta pesquisa a produtos comercialmente viáveis para assim auxiliar produtores rurais. Com este estudo queremos também mostrar que há como encurtar este caminho aplicando os princípios utilizados na indústria dentro de nosso processo de investigação.

**Palavras-chave:** Reconhecimento de eventos suspeitos em vídeo. Aprendizado não supervisionado. Vigilância rural.



## ABSTRACT

Recognizing suspicious events in videos is one of the biggest challenges in the field of computer vision. This work addresses the identification of abnormal situations in a scenario that has been little studied until then, the rural area. This scenario was selected due to the increasing crime rate in these areas and the dangers that crimes like these unleash to society. Although most of the methods proposed for recognizing abnormal events are based on supervised learning, there are some difficulties in using this type of learning in the surveillance task. Therefore, we propose the use of an event filter that receives trajectory *clustering* data from a genetic algorithm combined with the supervised object tracking and detection system *Yolo* for the recognition of abnormal situations in videos. A *dataset* for the chosen scenario was constructed using internet videos that proved to be good candidates for the study. Simulations of artificial trajectories were generated in order to validate the results with the real data generated. The results of this study were promising, despite the intensive processing of the genetic algorithm and the sensitivity of the tracker, the filter was able to delineate the abnormal events with both artificial and real data with satisfactory success. There is currently a great potential for boosting the results of this research into commercially viable products to help rural producers. With this study we also want to show that there is a way to shorten this path by applying the principles used in the industry within our research process.

**Key-words:** Recognition of suspicious events on video. Unsupervised learning. Rural surveillance.



## LISTA DE FIGURAS

Figura 1 – Processo Metodológico da Revisão. . . . .	19
Figura 2 – Comparação de Publicações por Ano. . . . .	20
Figura 3 – Métodos Identificados e suas Relações. . . . .	28
Figura 4 – Detecção de Objetos <i>Yolo</i> . . . . .	34
Figura 5 – Ciclo do Filtro de Kalman. . . . .	35
Figura 6 – Trajetória Bruta de uma Junção. . . . .	36
Figura 7 – Resultado do <i>Clustering</i> das trajetórias baseado em GPDfSC. . . . .	37
Figura 8 – Fluxograma Algoritmo Genético. . . . .	38
Figura 9 – Processo de Seleção/Criação de <i>Dataset</i> . . . . .	39
Figura 10 – Macroprocesso Metodológico. . . . .	41
Figura 11 – Estrutura do Repositório. . . . .	48
Figura 12 – Arquitetura do Repositório Central. . . . .	49
Figura 13 – Arquivo de Configuração. . . . .	50
Figura 14 – Detecção de Objetos. . . . .	51
Figura 15 – Sem Filtro de Kalman x Com Filtro de Kalman. . . . .	52
Figura 16 – <i>Clustering</i> e <i>Fitness</i> de 10 Números Sequenciais. . . . .	54
Figura 17 – Representação de Trajetórias Artificiais e os seus Valores Médios. . . . .	55
Figura 18 – <i>Clustering</i> e <i>Fitness</i> das Trajetórias Artificiais. . . . .	56
Figura 19 – <i>Clustering</i> das Trajetórias Individuais de cada Objeto Baseado nas Trajetórias Artificiais. . . . .	57
Figura 20 – Possíveis <i>Clusters</i> dos Dados das Trajetórias Artificiais Esperados. . . . .	58
Figura 21 – Identificação de Suspeitos com Base na Distância. . . . .	59
Figura 22 – <i>Clustering</i> de uma Trajetória Real vs Artificial. . . . .	62
Figura 23 – Gráfico da Trajetória Real 1. . . . .	63
Figura 24 – Gráfico da Trajetória Real 5. . . . .	63
Figura 25 – Gráfico da Trajetória Real 3. . . . .	64
Figura 26 – Gráfico da Trajetória Real 8. . . . .	64
Figura 27 – Gráfico da Trajetória Real 9. . . . .	64
Figura 28 – Gráfico da Trajetória Real 11. . . . .	65
Figura 29 – Resultados do Subconjunto com os Vídeos Reais. . . . .	65



## LISTA DE TABELAS

Tabela 1 – Relação de Métodos e Medidas Identificados. . . . .	25
Tabela 2 – Relação de Trabalhos Selecionados e Questões de Pesquisa. . . . .	26
Tabela 3 – Rótulos das Cores da Figura 3. . . . .	27
Tabela 4 – Critérios para Seleção dos Vídeos. . . . .	39
Tabela 5 – <i>Dataset</i> . . . . .	40
Tabela 6 – Formato dos Dados de Trajetórias. . . . .	44
Tabela 7 – Formato dos Dados de Interações. . . . .	44
Tabela 8 – Subconjunto de Dados Reais. . . . .	54
Tabela 9 – Teste de Funções de Dissimilaridade. . . . .	59
Tabela 10 – Resultados com Trajetórias Artificiais. . . . .	61



## SUMÁRIO

1	INTRODUÇÃO . . . . .	15
1.1	Motivação . . . . .	15
1.2	Objetivos . . . . .	17
1.3	Organização do Trabalho . . . . .	18
2	TRABALHOS RELACIONADOS . . . . .	19
2.1	Processo Metodológico da Revisão . . . . .	19
2.2	Aprendizado Supervisionado versus Não Supervisionado . . . . .	20
2.3	Revisão de Trabalhos . . . . .	21
2.3.1	Protocolo . . . . .	21
2.3.1.1	Objetivo e Escopo . . . . .	21
2.3.1.2	Questões de Pesquisa . . . . .	21
2.3.1.3	Processo de Pesquisa . . . . .	22
2.3.1.4	Estratégia de Extração das Informações . . . . .	23
2.3.1.5	Processo de Seleção . . . . .	23
2.3.1.6	Estratégia de Análise . . . . .	24
2.3.2	Resultados . . . . .	24
2.4	Considerações do Capítulo . . . . .	31
3	FUNDAMENTAÇÃO TEÓRICA . . . . .	33
3.1	Visão Computacional . . . . .	33
3.2	Aprendizado Supervisionado . . . . .	33
3.2.1	Detecção de Objetos . . . . .	34
3.3	Rastreamento de Objetos . . . . .	35
3.4	Trajectoria . . . . .	35
3.5	Aprendizado Não Supervisionado . . . . .	36
3.5.1	<i>Clusterização</i> . . . . .	36
3.6	Algoritmo Genético . . . . .	37
3.7	Arquitetura Flexível . . . . .	38
4	METODOLOGIA . . . . .	39
4.1	Definição do <i>Dataset</i> . . . . .	39
4.1.1	Definição do Subconjunto de Vídeos . . . . .	40
4.2	Construção da Proposta . . . . .	41
4.2.1	Detecção de Objetos . . . . .	42
4.2.1.1	Conservador de Id . . . . .	42
4.2.1.2	Cálculo de Coordenadas . . . . .	43
4.2.1.3	Extração das Coordenadas/Dados . . . . .	43
4.2.2	Reconhecimento de Eventos . . . . .	44

4.2.2.1	Análise do Comportamento Individual . . . . .	44
4.2.2.2	Interpretação do Comportamento do Grupo . . . . .	46
4.2.3	Definição da Arquitetura . . . . .	48
5	RESULTADOS . . . . .	51
5.1	Detecção de Objetos . . . . .	51
5.2	Reconhecimento de Eventos . . . . .	53
5.2.1	<i>Clustering</i> de Trajetórias . . . . .	53
5.2.2	Filtro de Interação . . . . .	58
6	CONSIDERAÇÕES FINAIS . . . . .	67
	REFERÊNCIAS . . . . .	69

## 1 INTRODUÇÃO

A detecção automática de eventos anormais em vídeos através de aprendizado de máquina é um tópico de grande relevância e interesse científico. Ao longo dos anos, muitas soluções já foram propostas, sendo a maioria baseadas em aprendizado supervisionado (QI; LIU; FU, 2020). Um dos maiores desafios deste tipo de abordagem consiste no grande número de amostras rotuladas necessárias para o treinamento do modelo em questão. O processo de construção de um conjunto de dados (*dataset*) de treino com características disponíveis que sejam adequadas e que consiga alcançar um resultado aceitável é consideravelmente complexo e trabalhoso (QI; LIU; FU, 2020). Em contrapartida, Kiran, Thomas e Parakkal (2018) apresentam uma visão global das técnicas de detecção de eventos anormais baseadas em métodos de reconhecimento de padrões não supervisionados já existentes para este mesmo fim, sendo que estas são comumente definidos com base em modelos estatísticos, os quais precisam ser definidos para que seja possível identificar o evento desejado.

Sendo assim, como um comportamento anormal deve ser definido em um vídeo? Esta é uma das principais questões que estamos investigando neste trabalho. Definitivamente não há uma resposta correta única para esta pergunta, pois estes comportamentos podem variar dependendo do cenário, podendo assim ser modelados de diversas maneiras. Não há uma definição da melhor modelagem para esses eventos, apenas consensos sobre alguns atributos úteis que servem para esta modelagem como, por exemplo, as trajetórias percorridas pelos elementos detectados na cena, consideradas informações cruciais que permitem a análise de eventos, principalmente na tarefa de vigilância automatizada (GHRAB; FENDRI; HAMMAMI, 2016). Outra característica importante é a formação de agrupamentos os quais, por sua vez, são os conjuntos de objetos detectados na cena, de modo que compreender suas propriedades e relações é essencial na interpretação de comportamentos (PALANISAMY; MANIKANDAN, 2017).

Neste trabalho propomos a combinação de análise de trajetórias com análise de grupo. Se compreendermos como cada grupo se comporta utilizando sua trajetória como parâmetro definidor, poderemos então interpretar o impacto e a relação entre os grupos existentes. Como a maioria das soluções encontradas trabalham na relação do comportamento entre seres da mesma espécie ou veículos, trazemos com este trabalho uma proposta de análise das relações suspeitas entre humano e animal num cenário rural utilizando uma abordagem de aprendizado não supervisionado.

### 1.1 Motivação

Na busca por soluções que realizem o reconhecimento de eventos em vídeo foram encontradas inúmeras soluções propostas com a utilização de métodos de aprendizado supervisionado. No entanto, quando o assunto é a utilização de métodos não supervisionados para este fim, o número de propostas diminui consideravelmente conforme apresentado

na figura 2.2. Ademais, dentre os trabalhos que utilizam métodos não supervisionados, não foi possível localizar nenhum que tratasse do cenário específico escolhido em nossa pesquisa e, mesmo dentre as propostas que utilizam métodos supervisionados, apenas 1,56% dos trabalhos foram direcionados a este cenário (WINCKLER; THIELO, 2020). Vale salientar que é importante preocupar-se com cenários que possuem algum tipo de demanda devido à limitação da relação entre a solução proposta e o cenário específico (WINCKLER; THIELO, 2020). Cada cenário possui elementos de interesse distintos, o que torna inviáveis determinadas reutilizações de abordagens.

A principal motivação para a escolha do cenário tratado neste estudo tem relação com a criminalidade na zona rural. Conforme apresentado por Ribeiro (2019) em seu estudo no período entre 2018 e 2019, pelo menos dois terços dos produtores rurais já foram vítimas de algum crime em função da atividade agropecuária e 51,5% não se sentem seguros durante a noite em sua própria propriedade.

O tópico da segurança rural tem ganhado cada vez mais força devido às inúmeras ocorrências de crimes em diferentes regiões do país nos últimos anos. Notícias recentes expõem a preocupação dos produtores rurais com isto (PÉTRIN, 2021). A conduta típica dos crimes já relatados consiste na subtração de animais de seu local de origem e depósito ou venda clandestina dos mesmos com a finalidade de produção e/ou comercialização (SILVA, 2016). Além disso, animais são abatidos e furtados independentemente do tamanho (PÉTRIN, 2021) e os crimes não se limitam apenas ao furto de bovinos, mas também de caprinos, suínos, equinos, entre outros animais (CNA, 2021).

Em 2016 o abigeato (furto de animal) foi reconhecido em uma qualificadora específica no artigo 155 do Código Penal, Lei n.º 13.330/2016, que tem como pena a reclusão de 2 a 5 anos se a subtração for de semovente domesticável de produção, ainda que abatido ou dividido em partes no local da subtração. Este é apenas um dos tantos crimes que resultam no uso e/ou comercialização do material furtado, de modo que o produto comercializado é a própria carne ou o animal vivo em si (SILVA, 2016).

De acordo com CNA (2021), devido ao desenvolvimento do Brasil nas atividades agropecuárias, aumento da produção e, conseqüentemente, maior riqueza no campo, um aumento da criminalidade também acabou afetando o setor. Não obstante os crimes na zona rural também aumentam em decorrência do menor policiamento nas zonas rurais. Além disso, este tipo de crime não afeta apenas o patrimônio dos pecuaristas, mas, também, o sistema tributário e a sociedade como um todo, sem contar com os vários problemas que estes crimes desencadeiam, oferecendo sérios riscos inclusive a saúde das pessoas que adquirem e consomem os produtos de procedência e qualidade duvidosas através da comercialização clandestina. Apesar da Lei n.º 13.330/2016 tratar, também, da fiscalização da procedência da carne vendida nos estabelecimentos e punir, igualmente, as pessoas que apoiam esta prática comprando destes estabelecimentos, a melhor alternativa continua sendo a prevenção e a adoção de cuidados para evitar esses prejuízos (SANTOS,

2021).

Infelizmente, no quesito de atuação da segurança pública, o meio rural ainda é uma área esquecida (CNA, 2021) e não há uma estrutura adequada para comunicação entre os produtores rurais e as autoridades responsáveis pela vigilância (PÉTRIN, 2021). Apesar deste esquecimento e dificuldades, algumas iniciativas estão sendo tomadas em diferentes regiões do país por órgãos interessados em preservar o agronegócio. Por exemplo, a Confederação da Agricultura e Pecuária do Brasil (CNA) promoveu em Brasília quatro painéis, entre 2019 e 2020, sobre segurança rural com especialistas das Polícias Militares de todo o Brasil, participa de debates sobre políticas públicas para melhorar as condições de segurança e patrulhamento no campo, dentre outros projetos e iniciativas (CNA, 2021). Em locais de estados como Santa Catarina e Paraná, foram implementados programas de patrulhamento rural com o uso de GPS a fim de facilitar a identificação das propriedades e prestar socorro (PÉTRIN, 2021). Em Santo Antônio da Patrulha, está prevista a criação de uma política pública municipal informatizada para cadastro de marcas e fiscalização, bem como a criação de novas categorias especializadas na repressão de crimes contra o patrimônio ocorridos principalmente em zonas rurais (FRANKEN, 2021).

Uma das grandes dificuldades no combate do abigeato, por exemplo, é que as propriedades são extensas e longe das cidades (PÉTRIN, 2021) e apesar da lei existir e punir devidamente os culpados, ela traz apenas uma falsa sensação de segurança pois existem muitos desafios no que tange a identificação do crime no momento exato e do possível responsável (SILVA, 2016), além de que é extremamente difícil conseguir encontrar os responsáveis por tais atos essencialmente por conta do atraso na reportagem do crime pela vítima, que muitas vezes só descobre o que aconteceu após terem passado horas do ocorrido.

Como vimos, já existem algumas medidas a se adotar para prevenir e frustrar a ação dos criminosos no campo e que a prevenção continua sendo a melhor opção. Com o auxílio da tecnologia, talvez, pudéssemos aumentar a sensação de segurança do produtor rural e desta forma fazer valer as leis que o protegem. Visa-se alcançar este objetivo por meio da utilização de dispositivos tecnológicos a fim de disponibilizar a vigilância automatizada dos animais nos momentos em que estes estivessem vulneráveis a ataques. Tendo isto em vista, selecionamos o cenário rural como ambiente para realizar nossos experimentos e focamos nossos esforços em reconhecer ocorrências suspeitas em atividades pecuárias.

## **1.2 Objetivos**

O objetivo geral deste trabalho consiste em apresentar uma proposta de método de detecção de eventos anormais em vídeos baseada em aprendizado não supervisionado, implementado utilizando uma arquitetura flexível para auxiliar os moradores de zona rural na vigilância de seus animais e de sua propriedade.

Para alcançar o objetivo geral, foram definidos os seguintes objetivos específicos:

- Realizar uma revisão de literatura para encontrar e analisar métodos já propostos para a detecção de eventos anormais em vídeos utilizando aprendizado não supervisionado;
- Encontrar e/ou construir um *dataset* de vídeos apropriado;
- Encontrar e/ou desenvolver um detector e rastreador de objetos em vídeos;
- Definir medidas e métodos a serem adotados no desenvolvimento da proposta;
- Promover uma prova de conceito da proposta;
- Definir a arquitetura mais adequada para o sistema de modo que esta seja de fácil manutenção e integração contínua.

### 1.3 Organização do Trabalho

Este documento está organizado da seguinte maneira:

- **(Capítulo 2)** Trabalhos Relacionados: Este capítulo apresenta a metodologia aplicada para a revisão de trabalhos, uma comparação quantitativa entre os tipos de aprendizado de máquina, bem como o protocolo da revisão, os resultados obtidos e as análises realizadas;
- **(Capítulo 3)** Referencial Teórico: Informações sobre os principais conceitos abordados neste trabalho;
- **(Capítulo 4)** Metodologia: Descrição da metodologia adotada para o desenvolvimento da proposta, as deduções, algoritmos e técnicas utilizadas para alcançar os objetivos deste trabalho;
- **(Capítulo 5)** Resultados: Apresenta os testes, validações e prova de conceito deste estudo;
- **(Capítulo 6)** Considerações Finais: Expõe considerações gerais sobre todo o processo, desde a relevância deste estudo, passando pela obtenção do *dataset* e sua qualidade, até a integração final do mesmo.

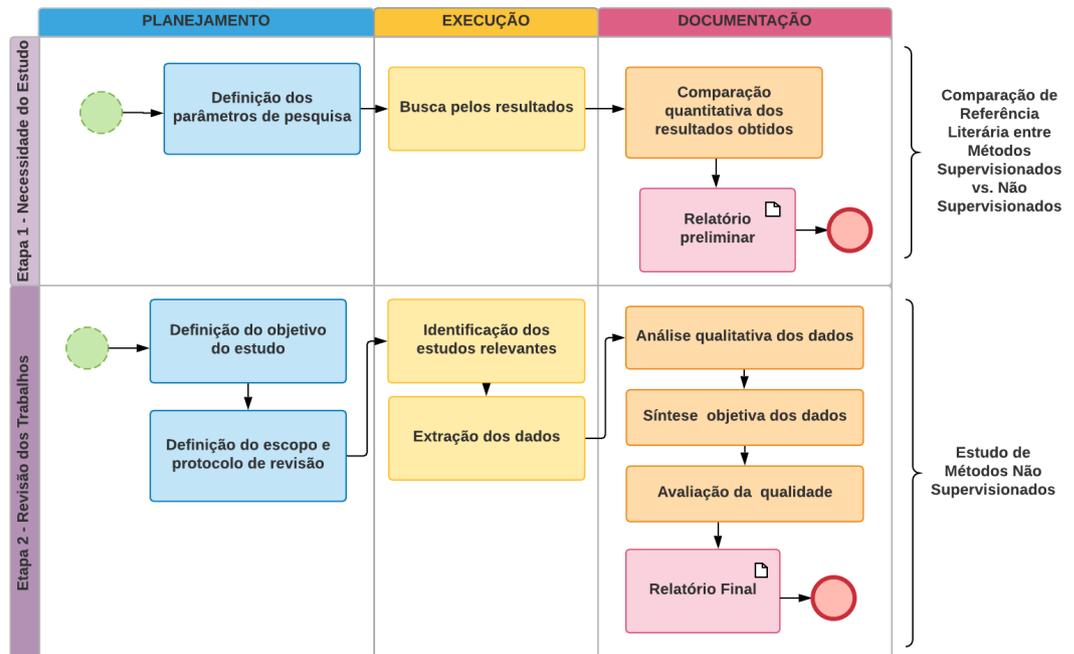
## 2 TRABALHOS RELACIONADOS

Neste Capítulo são apresentadas as etapas metodológicas executadas para seleção de trabalhos relacionados na Seção 2.1, os resultados preliminares com evidências da relevância desta pesquisa na Seção 2.2, o protocolo e os resultados da revisão na Seção 2.3 e, por fim, as considerações deste Capítulo na Seção 2.4.

### 2.1 Processo Metodológico da Revisão

O estudo de mapeamento sistemático identifica, seleciona e avalia criteriosamente a pesquisa a fim de responder a uma questão bem definida (BARN; BARAT; CLARK, 2017). O Mapeamento é construído sobre um processo estruturado e definido neste trabalho. Esse processo define as etapas necessárias para alcançar os objetivos do estudo. A Figura 1 mostra as etapas e as tarefas que foram realizadas.

Figura 1 – Processo Metodológico da Revisão.



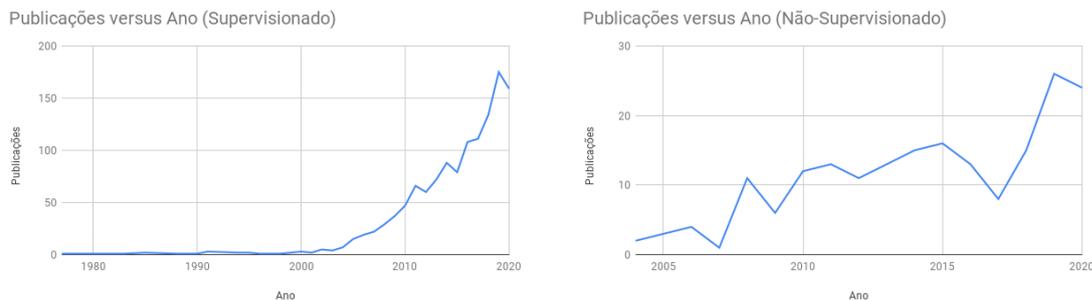
Fonte: Autor (2021).

O relatório preliminar definido na Figura 1 está apresentado na Seção 2.2, na qual são feitas apenas análises quantitativas, o relatório final definido na Figura 1 e o protocolo na Seção 2.3, onde os conceitos qualitativos são combinados com os quantitativos nas análises.

## 2.2 Aprendizado Supervisionado versus Não Supervisionado

Nesta Seção é apresentada uma comparação quantitativa entre os tipos de aprendizado de máquina de modo a prover embasamento teórico apontando a necessidade deste estudo. Foram realizadas duas buscas na base de dados da Scopus<sup>1</sup> por trabalhos que abordam a detecção de eventos suspeitos em vídeos, sendo uma listando apenas os que utilizam métodos não supervisionados e outra apenas os que não os utilizam. Logo de imediato é possível observar a disparidade de atenção dada aos tópicos ao longo dos anos, tendo sido encontrados 1290 resultados de publicações até 2020 na área de aprendizado supervisionado em comparação com apenas 196 com a utilização de aprendizado não supervisionado. A Figura 2 apresenta o número de publicações nas áreas ao longo dos anos.

Figura 2 – Comparação de Publicações por Ano.



Fonte: Autor (2021).

É possível observar que a abordagem de aprendizado não supervisionado para este fim recentemente tem ganhado força, tendo mais de 20 publicações por ano desde 2019. Mesmo se, quando comparado com o número de publicações de métodos supervisionados por ano, os métodos não supervisionados apresentem um número bem inferior, isso representa que há bastante espaço para exploração nesta área. No que diz respeito às publicações sobre o método supervisionado, observamos que desde 2016 esta área ganhou muita força, contando com mais de 100 publicações por ano e sua exploração vem de muito tempo antes. Se prestarmos mais atenção, em 2008 se iniciava um movimento maior neste tópico com a utilização de métodos supervisionados, começando a ganhar a presença que os métodos não supervisionados estão tendo atualmente. Se assumirmos que esse comportamento é uma tendência, pode-se dizer que provavelmente daqui a aproximadamente 12 anos haverá tanto interesse em métodos não supervisionados quanto há de métodos supervisionados nos dias de hoje.

Outro ponto que pode ter sido impactante na popularidade de métodos supervisionados foram as práticas de *deep learning* que ganharam um espaço muito grande nos

<sup>1</sup> SCOPUS : <<https://www.scopus.com>>

últimos anos.

## 2.3 Revisão de Trabalhos

Nesta Seção o protocolo e os resultados da revisão são apresentados. Esta revisão serve como referência para o desenvolvimento deste trabalho. Optou-se por não delimitar a busca apenas para soluções voltadas para o cenário que desejamos explorar neste estudo, tendo um foco maior em revisar quais as abordagens já foram exploradas para a detecção de anomalias em vídeos.

### 2.3.1 Protocolo

Nesta Subseção é apresentado o protocolo aplicado na revisão dos trabalhos relacionados. Este serve como um guia do estudo, a fim de direcioná-lo e trazer resultados objetivos que agreguem valor a esta pesquisa.

#### 2.3.1.1 Objetivo e Escopo

Tem-se o objetivo de apresentar a relação de trabalhos que utilizam métodos não supervisionados e suas conexões com o presente estudo. Utilizou-se o paradigma GQM (*Goal, Question, Metric*) (EUSGELD; FREILING; REUSSNER, 2008) para a definição do objetivo central desta revisão.

##### OBJETIVO DE ACORDO COM O PARADIGMA GQM:

*Com o objetivo de:* Identificar / Caracterizar

*No que tange:* A detecção de Eventos Suspeitos utilizando métodos de aprendizado não supervisionado

*Do ponto de vista de:* Pesquisador

*No contexto de:* Detecção de eventos suspeitos em vídeos de segurança

#### 2.3.1.2 Questões de Pesquisa

As questões foram definidas de acordo com o método de aprendizado de máquina adotado neste trabalho e suas características particulares.

QP1) Analisa o comportamento do grupo de elementos identificados?

QP2) Analisa o comportamento individual de elementos identificados?

QP3) Apresenta técnica de extração de características?

QP4) Utiliza algum método supervisionado de apoio (ex: para detecção de objetos de cena)?

QP5) Quais métodos utiliza?

Observação (QP5): Para mapeamento dos métodos utilizados em cada trabalho foram definidos os acrônimos MDA para Métodos de Detecção de Anomalia e MDT para Medida de Discriminação de Trajetória.

### 2.3.1.3 Processo de Pesquisa

A busca por trabalhos foi realizada apenas na base de dados literária Scopus devido a sua capacidade de unir outras base de dados na pesquisa, sendo encontrados trabalhos publicados em plataformas digitais distintas como a IEEE, ACM e SpringerLink, por exemplo.

Para a definição da *string* de busca foram definidos parâmetros essenciais para encontrar trabalhos relacionados utilizando apenas termos relacionados e operadores de comando em diferentes combinações a fim de obter resultados satisfatórios.

```
TITLE-ABS-KEY( ( "Anomalous"OR "Abnormality "OR "Abnormal") AND ( "Detection"OR "Recognition") AND ( "Action"OR "Behaviour"OR "Event") AND ( "Surveillance Videos"OR "Video") AND ("Clustering"OR "Cluster"OR "Unsupervised Method") ) AND ( LIMIT-TO ( PUBSTAGE,"final") ) AND ( LIMIT-TO ( LANGUAGE,"English") )
```

#### **Critérios de Inclusão:**

- CI1) Apresentar alguma técnica para interpretação de comportamento anômalo.
- CI2) Ter sido publicado em inglês.
- CI3) Utilizar vídeo como entrada do método proposto.
- CI4) Analisar comportamento de seres vivos.

#### **Critérios de Exclusão:**

- CE1) Utilizar algum método supervisionado para análise do comportamento.
- CE2) Apresentar método que classifica apenas imagens.
- CE3) O artigo estar incompleto ou ser um resumo.
- CE4) Utilizar sensores e/ou codificadores de imagem em hardware (ex: Inkjet-printing) para detecção de objetos e/ou identificação dos eventos.
- CE5) Texto não possuir autoria explícita (ex: resumo/apresentação de *proceedings*).

#### **Critérios de Avaliação da Qualidade:**

O objetivo da utilização destes critérios consiste em avaliar a relevância dos trabalhos selecionados para o estudo proposto e a sua capacidade em responder as questões de pesquisa definidas. Os critérios definidos e as possíveis avaliações estão definidos na lista a seguir:

**QA1)** Este artigo está conectado de maneira significativa à proposta que o presente trabalho traz?

S: Sim

P: Parcialmente

N: Não

**QA2)** Este artigo obteve algum resultado após aplicação do método proposto?

S: Sim

P: Parcialmente

N: Não

**QA3)** Utiliza trajetórias para análise do comportamento?

S: Sim

P: Parcialmente

N: Não

**QA4)** Utiliza alguma técnica de *clustering* para a identificação de padrões?

S: Sim

P: Parcialmente

N: Não

Cada um dos critérios citados é avaliado de acordo com a seguinte definição: Sim (S) = 1; Parcial (P) = 0,5; Não (N) = 0. Deste modo a pontuação total varia conforme as respostas das quatro perguntas, podendo resultar em: 0 a 1,0 (muito ruim); 1,5 ou 2,0 (regular); 2,5 ou 3,0 (bom); 3,5 ou 4,0 (muito bom).

#### 2.3.1.4 Estratégia de Extração das Informações

Foi extraído da base de dados da Scopus um arquivo com os dados de interesse dos trabalhos, sendo estes: Autores, Título, Ano da Publicação, *Abstract*, link de acesso e *Keywords*. Uma Tabela no *Excel* foi construída com estes dados e a partir deste e cada artigo foi buscado pelo link de acesso da Scopus e no Google Scholar<sup>2</sup> a fim de ter acesso ao texto de todos os trabalhos encontrados. Em seguida, duas colunas foram adicionadas nesta Tabela: Uma para mapear os métodos de *clustering* propostos e outra para mapear as medidas de dissimilaridade adotadas na diferenciação de dados dos *clusters*.

#### 2.3.1.5 Processo de Seleção

Após buscar por trabalhos relacionados na base de dados da Scopus, foram obtidos 196 resultados. Assim como apresentado na Seção 2.2, estes resultados foram extraídos conforme a definição na Subseção 2.3.1.4. Como os textos de alguns dos trabalhos não foram encontrados e alguns deles tinham acesso restrito de bibliotecas específicas, estes trabalhos foram removidos devido a não ser possível extrair as informações necessárias para esta revisão. Em seguida, nos trabalhos obtidos foram aplicados os seguintes filtros:

- Eliminação de Réplicas;
- Aplicação dos critérios de inclusão e exclusão a partir do *Abstract*;

<sup>2</sup> Google Scholar : <<https://scholar.google.com.br>>

- Aplicação dos critérios de inclusão e exclusão no corpo do texto;
- Avaliação da Qualidade: Somente trabalhos classificados como muito bom foram selecionados;

Após a aplicação dos critérios de inclusão e exclusão obtivemos 103 artigos, que foram filtrados novamente com a aplicação das questões de qualidade produzindo um total de 33 resultados.

### 2.3.1.6 Estratégia de Análise

Para a etapa de análise, os dados de interesse foram transcritos para a Tabela 2, na qual todas as questões de pesquisa foram respondidas e relacionadas com o respectivo trabalho, sendo a partir destas realizadas as análises.

Para a **(QP5)**, devido à sua abrangência, foi gerada a Tabela 1 para mapeamento dos métodos utilizados e seu respectivo tipo. Além disso, na Figura 3, é apresentado o grafo dos métodos mapeados a fim de conseguir obter todas as informações para análise do resultados.

### 2.3.2 Resultados

Nesta sub-seção são apresentadas as respostas às questões de pesquisa definidas de maneira objetiva e sistemática, bem como uma breve discussão referente a análise das respostas obtidas. Os trabalhos selecionados estão apresentados na Tabela 2 em relação às questões de pesquisa definidas. Na Tabela 1 foram mapeados os métodos identificados na revisão e classificados de acordo com abordagem proposta, sendo divididos em método de *clustering*, matemático/estatístico e *framework*/outros.

A fim de apresentar na Tabela 2 as respostas relacionadas a **(QP5)**, os métodos e medidas foram mapeados de acordo com os identificadores definidos na Tabela 1.

Com relação a **(QP1)** e **(QP2)**, é perceptível que a maioria dos trabalhos se preocupam em compreender o comportamento individual de cada elemento sem se preocupar com o todo. Ou seja, não verifica a interação dos objetos de cena entre si, apenas se preocupando em encontrar padrões anormais de objetos de forma isolada. Esta abordagem tem se mostrado muito eficaz na detecção de anomalias em vídeos baseando-se nas trajetórias, havendo um acervo bem grande de diferentes propostas a serem adotadas para este fim com resultados promissores. Como pode ser observado em Jiang, Wu e Katsaggelos (2008), que a partir de uma série temporal com a utilização de **HMM**, **DG** e **BA** separa os grupos de trajetórias em normais e anormais, trazendo em seu estudo uma abordagem prática para detectar anomalias. Assim como Jiang, Wu e Katsaggelos (2008), grande parte dos autores citados na Tabela 2 utilizam um princípio semelhante que se baseia em encontrar o padrão comum e chamá-lo de evento normal e a partir deste ponto identificar padrões que ocorrem com frequência inferior e rotulá-los como anormais.

Tabela 1 – Relação de Métodos Identificados.

ID	Clustering	ID	Estatístico/Matemático	ID	Framework/Outros
(K-M)	<i>K-means</i>	(HMM)	<i>Hidden Markov Models</i>	(FFM)	<i>Force Field Model</i>
(K-NN)	<i>K-Nearest Neighbor</i>	(NB)	<i>Naive Bayes</i>	(SRM)	<i>Semantic Region Model Generation</i>
(SC)	<i>Spectral</i>	(BA)	<i>Bayesiana</i>	(SIFT)	<i>Scale Invariant Feature Transform</i>
(HC)	<i>Hyperspherical</i>	(ED)	<i>Euclidean Distance</i>	(TF)	<i>Third Feature</i>
(MSC)	<i>Mean-Shift</i>	(LCSS)	<i>Longest Common Subsequences</i>	(CDG)	<i>Conflict Descriptor of Groups</i>
(HC)	<i>Hierarchical</i>	(DBA)	<i>Distance-Based Approach</i>	(EF)	<i>Evaluation Function</i>
(KCH)	<i>Kth Cluster Histogram</i>	(LLE)	<i>Locally Linear Embedding</i>	(MLP)	<i>Severity Categorization with Multiphase Linear Regression</i>
(AMC)	<i>Adjacency Matrix-based</i>	(JSD)	<i>JENSEN-SHANNON DIVERGENCE</i>	(IB)	<i>INFTIFY Belonging to Sparse Regions</i>
(SOM)	<i>Self-Organizing Map</i>	(LP)	<i>Labelled with Probabilistic Information</i>	(DGS)	<i>DEPTH GREEDY SEARCH</i>
(DC)	<i>DBSCAN</i>	(SM)	<i>Sample Method</i>	(PWD)	<i>Pair-Wise Distances</i>
(HLF)	<i>High-Level Featuring-Based</i>			(DTW)	<i>Dynamic Time Warping</i>
(SL)	<i>Single-Link</i>	(E1)	<i>Equation 1</i>	(DP)	<i>Douglas-Peucker</i>
(GK)	<i>Global Kernel</i>	(EYM)	<i>Eps. and MinPts.</i>	(HF)	<i>Haversine Formula</i>
(AP)	<i>Affinity Propagation</i>	(MoG)	<i>Mixture of Gaussians</i>	(PVF)	<i>Pos-VDMMLD Feature</i>
(SNE)	<i>Stochastic Neighbour Embedding</i>	(HD)	<i>Hausdorff Distance</i>	(NTS)	<i>Novel Three Step Search</i>
(FR)	<i>Feature Representation</i>	(DG)	<i>Discrete Gaussian</i>	(DB)	<i>DaviesBouldin</i>
(NM)	<i>Novel Method</i>	(CA)	<i>Chebyshev Approximations</i>	(RA)	<i>Radius Approximation</i>
(FCM)	<i>Fuzzy C-Means</i>	(PA)	<i>Polynomials Approximations</i>	(NSMK)	<i>Novel Similarity Metric Based on Kernels</i>
(AH)	<i>Agglomerative Hierarchical</i>	(DFT)	<i>Discrete Fourier Transform</i>	(RTH)	<i>Partition-Trajectory Histogram</i>
(BC)	<i>Behavior</i>	(EM)	<i>Expectation Maximization</i>	(ODM)	<i>Outlier Detection Method</i>
		(MC)	<i>Median Cardinality</i>	(LK)	<i>Lucas-Kanade</i>
		(DD)	<i>Dirac Delta</i>	(FGA)	<i>Fast Global Alignment Kernel</i>
		(PM)	<i>Probabilistically Modeled</i>	(DMA)	<i>Distance Disparity Matrix Arrangement</i>
				(HT)	<i>Histogram Transformation</i>
				(HE)	<i>Histograms with Eqs.</i>

Fonte: Autor (2021).

Tabela 2 – Relação de Trabalhos Seleccionados e Questões de Pesquisa.

Referência	QP1	QP2	QP3	QP4	QP5(MDA)	QP5(MDT)
(NAFTEL; KHALID, 2005)		✓			K-NN	CA, PA, ED
(NAFTEL; KHALID, 2006b)		✓			SOM	CA, PA
(PICIARELLI; FORESTI, 2006)		✓			PM	ED, NM
(NAFTEL; KHALID, 2006a)		✓			SOM, K-NN	ED, DFT, CA, PA
(SWEARS; HOOGS; PERERA, 2008)		✓		✓	HC, HMM	HMM
(JIANG; WU; KATSAGGELOS, 2008)		✓			HMM, DGS	BA
(ANJUM; CAVALLARO, 2008)		✓	✓		DBA, IB	MSC
(GAO et al., 2008)					SRM	ED, HD
(CHEN; HUANG, 2010)	✓				AMC	LK, ED
(ESPINOSA-ISIDRÓN; GARCÍA-REYES, 2010)		✓			SL	NM
(AKÖZ; KARSLIGIL, 2010b)	✓		✓		MLP, BA	HMM, MOG, EM
(CHEN; HUANG, 2011)	✓		✓		AMC, FFM	LK
(MORRIS; TRIVEDI, 2011)		✓	✓	✓	HMM, BA	GMM, EM, LCSS, SC
(PALAZZO; SPAMPINATO; BEYAN, 2012)	✓		✓	✓	K-M, ED	EM, DP
(ZHANG et al., 2013)	✓		✓	✓	HMM	NM
(BRUN; SAGGESE; VENTO, 2013)		✓			K-M, GK	FGA, NSMK
(BEYAN; FISHER, 2013)		✓	✓		AP, ED	NM, ED, MC
(CUI; LIU; XING, 2014)		✓			KCH	FCM, HT, HE, DD, DG
(MOHAMED; SAUNIER, 2015)		✓			MSC	LCSS
(FEIZI; AGHAGOLZADEH; SEYEDARABI, 2015)		✓	✓	✓	SM	BC, SC, HMM
(YANG et al., 2016)		✓	✓	✓	HC, K-NN	FR
(FERNANDO et al., 2016)		✓		✓	K-M, SC	SC, DMA
(GHRAB; FENDRI; HAMMAMI, 2016)		✓	✓		SIFT, LCSS	LCSS, AH
(ZANG et al., 2016)		✓			K-M	NTS, RTH
(RANJITH; ATHANESIOUS; VAIDEHI, 2016)		✓	✓		DC	DC, EYM
(SAINI et al., 2017)		✓			HLF	K-M, ED
(XU et al., 2018b)		✓	✓	✓	EF	K-M, E1, DB
(XU et al., 2018a)		✓			CDG	ED, PVF, RA
(QAROUT; LOWE, 2019)		✓			LLE, JSD	HF
(PUDASAINI; ABHARI, 2019)		✓		✓	SC, NB	ED, DTW, SC
(ATHANESIOUS et al., 2019)	✓			✓	BA	SC, PWD, DTW
(FEIZI, 2020)		✓		✓	SC, TF	SC
(ATHANESIOUS et al., 2020)	✓				K-M, K-NN	HLF

Fonte: Autor (2021).

Entretanto, conforme Chen e Huang (2010) e Chen e Huang (2011) apenas a relação de um evento isolado pode não ser suficiente uma vez que os eventos induzidos pelas interações entre elementos de cena também são extremamente importantes, uma vez que comportamentos anormais possam ter certa influência em determinadas situações e estes resultariam ou representariam determinados danos específicos. Em seu estudo Chen e Huang (2011) comprovou a eficácia de sua afirmação mostrando que com sua abordagem utilizando *Adjacency Matrix-based*, *Force Field Model* e *Lucas-Kanade* é possível detectar eventos anômalos em ambientes descontrolados. Em Athanesious et al. (2019), Zhang et al. (2013), Athanesious et al. (2020) e Aköz e Karsligil (2010b) tem-se a ideia de analisar comportamento de grupos conforme o contexto, como uma combinação do todo pode tra-

zer o resultado de comportamento de grupo e especificamente Athanesious et al. (2019) em seu estudo traz o conceito da visão coletiva das trajetórias anormais. Já Palazzo, Spampinato e Beyan (2012) apresenta em sua obra a possibilidade de compreender interações entre trajetórias.

Na Tabela 2 pode-se verificar que alguns trabalhos apresentam estratégias de extração de dados (**QP3**) baseado em aprendizado não supervisionado como Beyan e Fisher (2013), Morris e Trivedi (2011), Aköz e Karşligil (2010a), Ranjith, Athanesious e Vaidehi (2016) e Fendri, Ghrab e Hammami (2017) enquanto outras utilizam aprendizado supervisionado (**QP4**) como Palazzo, Spampinato e Beyan (2012), Ghrab, Fendri e Hammami (2016), Zhang et al. (2013), Yang et al. (2016), Feizi, Aghagolzadeh e Seyedarabi (2015) e Xu et al. (2018b). É possível observar que os trabalhos que satisfazem a (**QP3**) trazem um esclarecimento maior da organização dos dados de entrada dos algoritmos além de contribuírem significativamente com estudos como este. No que diz respeito ao tipo de aprendizado escolhido em cada estudo, tem-se um empate pois ambos geram bons conjuntos de entrada com confiabilidade relativamente alta. Não há indícios de abordagens melhores ou mais adequadas para tal escolha, a qual é totalmente dependente do objetivo que se deseja alcançar.

Ainda com relação a (**QP4**), alguns estudos utilizam métodos supervisionados em alguma etapa específica que antecede a classificação de eventos anormais no apoio da obtenção de sua solução final, como Pudasaini e Abhari (2019) que os utiliza para distinção de padrões do conjunto de dados.

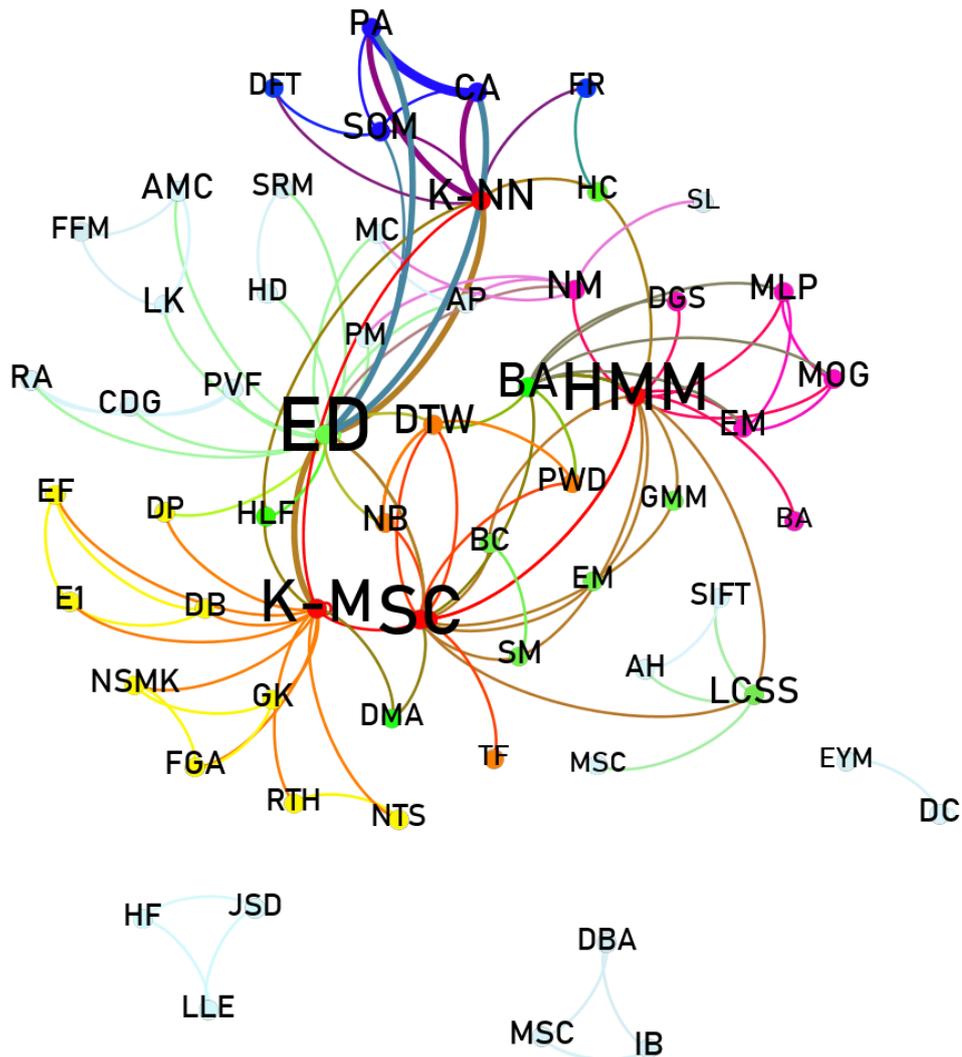
Como pode-se observar na Tabela 2 no que tange a (**QP5**) são exploradas diversas combinações de métodos. No entanto existem métodos que são utilizados com maior frequência nas soluções. Para uma melhor análise foi construído um grafo dos métodos (nós) e suas relações (arestas), apresentado na Figura 3, classificando os grupos de acordo com os métodos mais utilizados conforme a Tabela 3. Dentre os métodos observados na Tabela 2 quatro principais métodos foram destacados devido à quantidade de ocorrências nos trabalhos revisados e chamados na Tabela 3 de métodos raiz utilizados para classificação.

Tabela 3 – Rótulos das Cores na Figura 3.

	Métodos raiz que se relacionam diretamente entre si.
	Métodos que tem relação com mais do que um dos métodos raiz.
	Métodos que se relacionam diretamente com K-Means.
	Métodos que se relacionam diretamente com K-NN.
	Métodos que se relacionam diretamente com SC.
	Métodos que se relacionam diretamente com HMM.
	Métodos que não se relacionam diretamente com os métodos raiz.

Fonte: Autor (2021).

Figura 3 – Métodos Identificados e suas Relações.



Fonte: Autor (2021).

Na Figura 3 podemos observar com uma maior clareza os métodos que mais se destacam nos trabalhos relacionados e também observar que todos os quatro métodos que se destacam se relacionam com pelo menos um outro método do mesmo grupo, assim como inúmeras medidas de dissimilaridade também são compartilhadas entre eles. Outro fato interessante diz respeito ao número de vezes em que estas medidas são utilizadas junto com cada método, sendo que poucas são utilizadas mais do que uma vez numa mesma combinação. Além disso, existem alguns grupos isolados que podem ser observados, em grande parte compostos pelos métodos que cada autor desenvolveu em seu próprio estudo. Dentre os métodos investigados, os que mais se destacaram foram:

**(HMM)** *Hidden Markov Models* é um modelo probabilístico em que o sistema modelado é assumido como um processo de Markov com parâmetros desconhecidos, ideal

para séries temporais nas quais seja possível determinar os parâmetros ocultos a partir dos parâmetros observáveis. Os parâmetros extraídos do modelo podem então ser usados para realizar novas análises, por exemplo para aplicações de reconhecimento de padrões (EDDY, 1996). Por exemplo, Zhang et al. (2013) modela sequências de vídeos com estados e valores de observação. Assim, com base na aceleração, os movimentos suspeitos são distinguidos e a detecção de anomalia é realizada. Em seu trabalho Morris e Trivedi (2011) representa cada atividade usando um **HMM** pois ele lida naturalmente com a normalização do tempo e ainda cita que, devido à simplicidade de treinamento e da avaliação, este modelo é ideal para a implementação em tempo real. Em contrapartida Athanesious et al. (2019) observa que a performance do HMM depende totalmente do número de estados escondidos, o que faz com que o mesmo apresente desempenho satisfatório nos dados treinados, mas um desempenho insuficiente quando aplicado na prática.

(**K-M**) *K-Means* é um dos algoritmos de aprendizagem não supervisionados mais simples e conhecidos que pode ser útil na detecção de dados anormais e análises de conjuntos de dados. Xu et al. (2018b) utiliza em seu estudo esta abordagem para detectar os possíveis caminhos na cena, e assim identifica as trajetórias que não se encaixam indicando estas como anormais. Uma grande desvantagem observada por Xu et al. (2018b) consiste no número de *clusters*,  $k$ , que precisa ser definido manualmente e neste cenário é algo praticamente inviável. Para superar este desafio, Xu et al. (2018b) implementou uma abordagem iterativa para tentar descobrir o melhor  $k$  em um determinado cenário. O sistema não precisa anotar rótulos e pode ser adaptável ao aspecto temporal dos dados, superando assim alguns desafios apontados por Athanesious et al. (2019) em seu estudo. Os resultados do experimento de Xu et al. (2018b) mostram que a abordagem proposta pode ter sucesso em detectar e visualizar trajetórias anormais.

Ghrab, Fendri e Hammami (2016) também utiliza **K-M** para agrupar eventos de dados normais. O agrupamento simples e eficiente é usado para estudar o fluxo de movimento normal durante o treinamento. O resultado experimental que conta com a utilização do algoritmo **K-NN** para a detecção de anomalias mostra que a abordagem proposta está efetivamente detectando anomalias e é capaz de atingir resultados melhores do que as técnicas de última geração.

Por outro lado, Athanesious et al. (2019) defende que a técnica de *Spectral Clustering* é mais eficiente do que os **K-M** tradicionais que agrupam dados baseados no *cluster* de grafos em vez de espaço de dados nativo. E ainda considera que técnicas que utilizam várias medidas de distância para estimar a instância limiar, como o **GMM**, dependem totalmente da instância limiar, que tem um custo computacionalmente alto. Além inferir em seu estudo que os esquemas tradicionais de detecção de anomalias que utilizam abordagens estatísticas e probabilísticas não conseguem enfrentar determinados desafios como lidar com diferentes comprimentos das trajetórias, velocidades diferentes, indisponibilidade de dados rotulados e ruídos dos dados. Por estes motivos os autores utilizam

o GPDI com técnica de *Spectral Clustering* para agrupamento em seu artigo juntamente com **(BA)** para a detecção de anomalias. A teoria da decisão *Bayesiana* **(BA)** é uma abordagem estatística usada para classificação com base na probabilidade máxima de dados, que por sua vez funciona bem para dados contínuos e de séries temporais. Com aplicação destes ele reduziu drasticamente o tempo de execução na dedução de anomalias e mostrou que a regra de decisão *Bayesiana* foi eficiente o suficiente para reconhecer as anomalias com falsos positivos insignificantes, além de provar que sua proposta é viável para implantação em tempo real.

**(SC)** *Spectral Clustering* é um tipo de aprendizado não supervisionado utilizado para agrupar um conjunto de dados não rotulados. Este agrupa os dados utilizando uso do espectro da matriz de similaridade dos dados para realizar a redução da dimensionalidade antes de agrupá-los em menos dimensões. O objetivo do *Spectral Clustering* é agrupar dados conectados, mas não necessariamente compactos ou agrupados dentro de limites convexos (PUDASAINI; ABHARI, 2019). Em seu estudo (MORRIS; TRIVEDI, 2011) define o agrupamento espectral como uma técnica popular devido sua acurácia em comparação com os algoritmos mais tradicionais de *clustering*. Os métodos espectrais não fazem nenhuma soma sobre a distribuição de pontos de dados e, em vez disso, se resume à decomposição dos autovalores de uma matriz de similaridade que se aproxima de uma partição numérica ideal (MORRIS; TRIVEDI, 2011).

Em contrapartida, as considerações realizadas por Pudasaini e Abhari (2019) em seus resultados experimentais mostraram que *Naïve Bayes* é melhor do que *Spectral Clustering* para classificação de trajetórias, uma vez que os dados de entrada do *Spectral Clustering* são rotulados para o seu treinamento. Ademais, o *Naïve Bayes* se mostrou uma ferramenta efetiva para classificar o movimento de objetos.

Ranjith, Athanesious e Vaidehi (2016) cita que Epsilon ( $\epsilon$ ) e Pontos Mínimos (MinPts) são os parâmetros essenciais para calcular dinamicamente a soma dos *clusters* para um ponto de dados. Recentemente, Athanesious et al. (2020) propôs a utilização destes dois parâmetros através da seleção randômica destes com o método *Density Based Spatial Clustering of Applications with Noise (DBSCAN)* para *clustering* de trajetórias. Com isso é possível observar que os métodos mais tradicionais da literatura continuam sendo os mais efetivos e como mostra a Figura 3 nem sempre são os mais utilizados, tendo voltado a ser explorados em estudos mais recentes.

**(K-NN)** *K-Nearest Neighbours* é um método de classificação não paramétrico utilizado para classificação e regressão. Ghrab, Fendri e Hammami (2016) identifica os eventos anormais com esta técnica na fase de testes de seu estudo. Naftel e Khalid (2005) também utiliza um classificador k-NN ( $k=1$ ) para classificar todos os espectros de instâncias do conjunto de testes e gerar a precisão geral da classificação. Isso envolve o treinamento do classificador em todas as trajetórias rotuladas além da única trajetória a ser testada, corroborando assim com a visão crítica de Athanesious et al. (2019) para

métodos como este.

Um fato interessante a se considerar é que, dentre todos os trabalhos selecionados, apenas Xu et al. (2018b) utiliza o índice *Davies Bouldin*, que nada mais é do que uma métrica para avaliar algoritmos de agrupamento. Este índice pode auxiliar na coesão dos resultados devido ao seu caráter sistemático e *background* literário.

Com relação aos métodos mais utilizados para distinção dos dados tem-se em disparado a **(ED)** *Euclidean Distance* como podemos observar na Figura 3. Esta medida representa a distância entre dois pontos comuns. É importante destacar que a utilização desta medida para aproximação de pontos não é ideal para aproximações deste tipo. O problema é que ela pressupõe que ambas as trajetórias possuem o mesmo comprimento e este é um cenário pouco provável em situações como as que tratamos neste trabalho devido a variações de velocidade e oclusão dos elementos. Outro problema é que ele não pode lidar com a mudança de horário local. No entanto, é simples de implementar devido seu caráter visual (ESPINOSA-ISIDRÓN; GARCÍA-REYES, 2010). Na maioria dos casos observados nesta revisão, esta medida é combinada com outras técnicas. Por sua vez, as relações que mais chamam atenção em nosso grafo são:

**(PA)** *Polynomials Approximations* se preocupa com como os valores podem ser melhor aproximadas com funções mais simples, e com caracterização quantitativa dos erros introduzidos. O polinômio em si é escolhido de forma a minimizar o erro. Ou seja, o objetivo é minimizar o valor máximo de  $x$ , onde  $P(x)$  é o polinômio aproximador,  $f(x)$  é a função real, e o  $x$  varia ao longo do intervalo escolhido (TIMAN, 2014).

**(CA)** *Chebyshev Approximations*, que nada mais são do que herdeiras das **(PA)**, definidas como uma sequência de polinômios ortogonais, relacionados com a fórmula de Moivre, adquiridos de forma recursiva. São importantes na teoria da aproximação porque as raízes dos polinômios de primeira ordem podem ser utilizados na interpolação polinomial. O resultado da interpolação minimiza o problema do fenômeno de Runge e fornece a melhor aproximação de uma função contínua que obedece à norma do supremo. Sendo semelhante às análises de Fourier, usando os polinômios de Chebyshev ao invés das funções trigonométricas usuais (BOYD, 2001).

Estas duas abordagens se relacionem muitas vezes no grafo devido à sua correlação natural e são os únicos que se conectam mais do que uma vez com *Euclidean Distance*. Sendo assim, observa-se que mesmo sendo abordagens puramente matemáticas e relativamente antigas são as que mais possuem referências sobre sua capacidade na resolução do problema em questão, conforme observado nos estudos em Naftel e Khalid (2005), Naftel e Khalid (2006b) e Naftel e Khalid (2006a).

## 2.4 Considerações do Capítulo

Através da análise dos trabalhos relacionados, foi possível identificar as diversas abordagens que podem ser combinadas para chegar ao mesmo fim. Além de ter sido de

grande valor e apoio à fase de desenvolvimento e construção do referencial teórico deste trabalho.

Também foi possível perceber que existem muitas contradições no que diz respeito aos métodos e medidas mais adequados para detecção de anomalias em vídeos, em alguns casos é possível verificar contradições do próprio autor entre suas publicações. A razão para tal disparidade de opiniões pode ser dada pelos diferentes cenários abordados em cada trabalho, que variam entre comportamento de multidões, comportamento de peixes e análises de trânsito, bem como pelo simples fato de pontos de vistas, concepções diferentes e evolução dos métodos em si.

### 3 FUNDAMENTAÇÃO TEÓRICA

Neste Capítulo são apresentados de forma objetiva e direcionada os principais conceitos para o entendimento deste trabalho, começando pelo conceito de visão computacional, apresentado na seção 3.1. Sendo seguido pela definição de aprendizado supervisionado na seção 3.2, na qual os conceitos de detecção (subseção 3.2.1) e rastreamento de objetos (subseção 3.3) são apresentados com as devidas especificações dos métodos selecionados no desenvolvimento. Em seguida a definição de trajetória é apresentada na seção 3.4, a de aprendizado não supervisionado na seção 3.5 e o respectivo método relacionado (*Clustering*) em sua subseção 3.5.1. Por fim é descrito o conceito de algoritmos genéticos na seção 3.6 e a definição global do que arquitetura flexível representa na seção 3.7.

#### 3.1 Visão Computacional

A Visão Computacional é a sub-área da Ciência da Computação na qual máquinas são treinadas para serem capazes de perceber o mundo de maneira semelhante à percepção humana, de modo que a máquina seja capaz de entender a imagem como um todo e utilizar as informações obtidas para a tomada de decisões (SZELISKI, 2010). Pode ser definida então como um conjunto de técnicas que tem como objetivo o processamento de dados para transformação do dado da imagem em uma nova representação compreendida automaticamente pela máquina (GONZALEZ; WOODS, 2000). Apesar dos avanços nos últimos anos esta tarefa continua sendo uma das mais desafiadoras da computação devido à dificuldade de obter uma boa representação do ambiente que desejamos observar (SZELISKI, 2010).

No contexto no qual estamos inseridos neste trabalho se faz necessário que a máquina seja capaz de compreender o que está ocorrendo em sequências de imagens para que assim seja possível realizar análises. Existem algumas abordagens de aprendizado de máquina que apoiam essa compreensão, sendo estas: Aprendizado supervisionado e aprendizado não supervisionado.

#### 3.2 Aprendizado Supervisionado

Métodos de aprendizado supervisionado utilizam um conjunto de exemplos de treinamento rotulados, onde o rótulo de cada classe é conhecido e associado à respectiva amostra contida no conjunto de dados. Para rótulos de classe discretos tem-se o conceito de classificação e, para valores contínuos, regressão. Em geral, o objetivo deste tipo de aprendizado consiste em construir um classificador que possa determinar corretamente a classe de novos exemplos que não foram rotulados, ou seja, amostras diferentes das contidas no conjunto de treino (MONARD; BARANAUSKAS, 2003).

Um exemplo de aprendizado supervisionado utilizado no domínio deste estudo é a detecção de objetos, que diferente da classificação precisa, além de classificar o objeto,

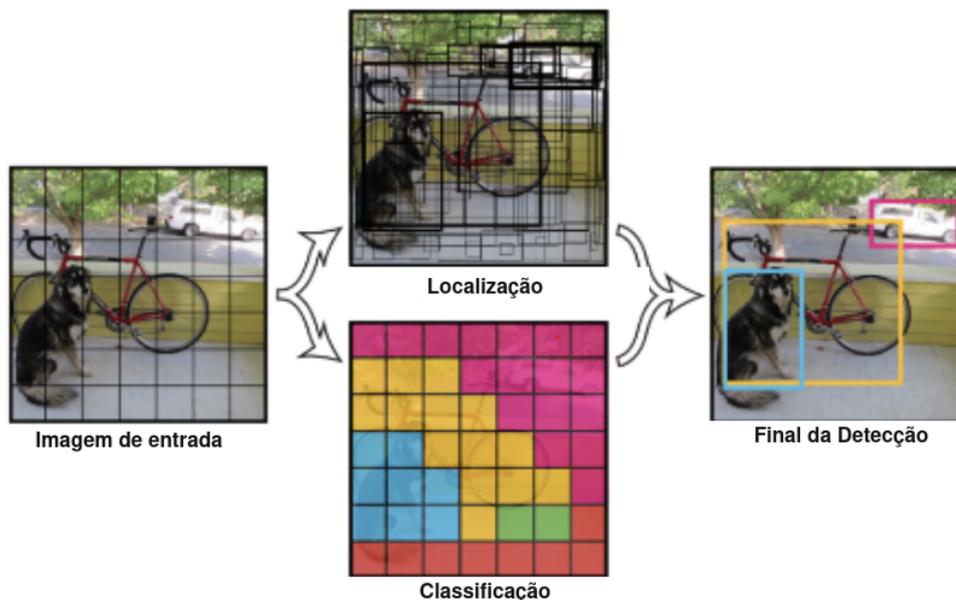
identificar a localização deste na imagem (ALVES, 2020).

### 3.2.1 Detecção de Objetos

A detecção de objetos é uma prática clássica de visão computacional, presente em aplicações de vigilância automatizada, análises de imagens médicas, dentre outras. Como citado na seção 3.2, a detecção tem como base a classificação e identificação do objeto em uma imagem. Sendo assim, a capacidade de prever a localização de um objeto na imagem juntamente com a classe deste (para múltiplas classes de objetos) é o que chamamos de detecção de objetos (ALVES, 2020).

A técnica denominada *Yolo (You Only Look Once)* é considerada estado da arte na área de detecção de objetos em tempo real (ALVES, 2020). Isto se dá devido sua grande popularidade e conceito inovador, que diferente abordagens anteriores, como R-CNN (*Region Based Convolutional Neural Networks*) e ou Faster R-CNN, ele olha apenas uma vez para a imagem e envia para a rede neural, aplicando a rede neural à imagem completa enquanto outros sistemas aplicam o modelo em vários locais de uma imagem (REDMON; FARHADI, 2018). Este fator contribui para que a velocidade da detecção seja muito superior as outras sem perder em acurácia (ALVES, 2020). Na Figura 4 podemos ver a detecção de objetos realizada pelo *Yolo*.

Figura 4 – Detecção de Objetos *Yolo*.



Fonte: (REDMON; FARHADI, 2018).

Como queremos trabalhar com vídeos neste estudo além de detectar os objetos na imagem precisaremos rastreá-los nas imagens subsequentes.

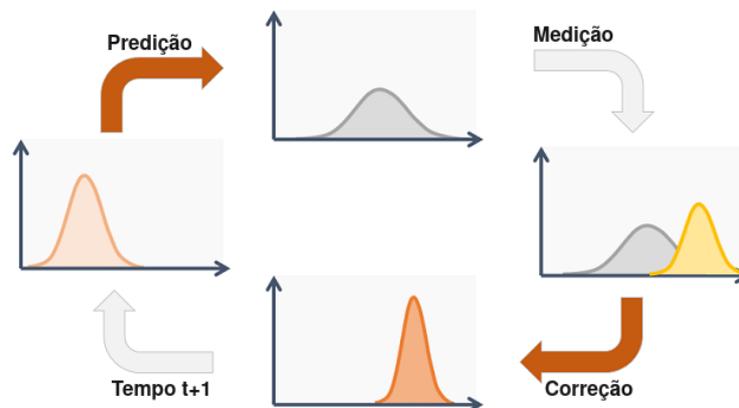
### 3.3 Rastreamento de Objetos

É importante ressaltar que há uma diferença entre detecção e rastreamento. Com a detecção somos capazes de localizar e classificar os objetos que compõem uma imagem, já o rastreamento nos dá informações sobre a localização desses objetos ao longo do vídeo (em uma sequência de imagens), ou seja, ao rastrear um objeto sabemos que o objeto identificado nos *frames* subsequentes provavelmente será o mesmo do início (MAIYA, 2019). Para tal utilizamos identificadores únicos (ids) referentes a cada objeto detectado e, também, as características espaço temporais dos objetos para prever seu Id (MAIYA, 2019).

De encontro com essa necessidade, o filtro de Kalman se mostra um bom candidato pois utiliza as detecções disponíveis e previsões anteriores para chegar a um melhor palpite do estado atual (MAIYA, 2019). Contando as etapas de predição e correção sobre as estimativas de processos gaussianos (SOARES; THIELO, 2020).

O filtro faz basicamente a propagação e atualização de gaussianas e atualização de suas covariâncias, prevendo inicialmente o próximo estado a partir da transição de estado obtida. Logo, se aplicável, as informações de medição ruidosas são incorporadas na fase de correção e o processo se repete conforme ilustrado na Figura 5 (JURIÉ, 2015).

Figura 5 – Ciclo do Filtro de Kalman.



Fonte: (JURIÉ, 2015).

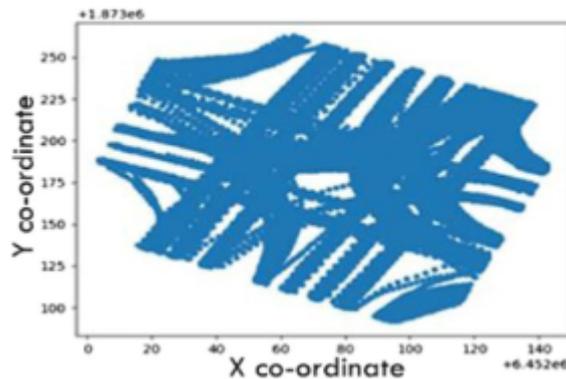
### 3.4 Trajetória

A trajetória de objetos é um dos parâmetros mais populares e amplamente utilizada em várias aplicações de vigilância, definida pela dinâmica de movimento de objetos. Podendo ser obtida a partir de vídeos utilizando *optical flow*, *dense trajectory*, *feature tracking* e *multi object tracking* (AHMED et al., 2018).

O rastreamento de objetos e a compreensão do comportamento é uma tarefa complexa e desafiadora que vem sendo amplamente estudada desde 2007. O objetivo principal

desses algoritmos é identificar objetos em movimento no domínio temporal e compreender seu comportamento através deste. Trajetórias podem ser utilizadas para vigilância visual automática, gerenciamento de tráfego, detecção de atividade suspeita, análise de vídeo esportivo, sumarização de vídeo, geração de sinopse, descritor de vídeo em texto, etc (AHMED et al., 2018). A Figura 6 apresenta um exemplo de rastreamento de várias trajetórias ao longo do tempo.

Figura 6 – Trajetória Bruta de uma Junção.



Fonte: (ATHANESIOUS et al., 2019).

### 3.5 Aprendizado Não Supervisionado

O aprendizado não supervisionado, diferente do supervisionado, parte do princípio da não utilização de dados pré rotulados, dispensando assim longos volumes de conjuntos de exemplos rotulados e treinos extensivos (Sodemann; Ross; Borghetti, 2012). Baseando-se em princípios matemáticos e probabilísticos para classificação dos dados de modo que a máquina sozinha seja capaz de compreender o padrão esperado.

No contexto em que deseja-se aplicá-lo normalmente a frequência de ocorrências é o parâmetro utilizado para diferenciação de anormalidades. Xiang e Gong (2005) observaram em seu estudo que métodos não supervisionados possuem um desempenho melhor do que métodos supervisionados para este fim. O método não supervisionado mais utilizado para tal é a *clusterização*.

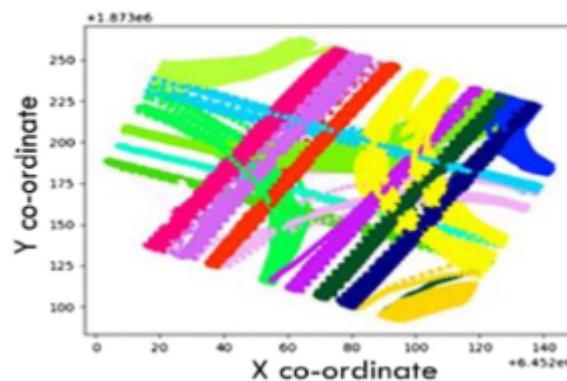
#### 3.5.1 Clusterização

A *clusterização* consiste em um método não supervisionado que cria agrupamentos de dados baseado em similaridade, os quais podem ser divididos baseados nos mais diversos padrões. Qualquer tipo de dado, desde que seja quantificável pode ser *clusterizado*. Esta técnica baseia-se em aproximações de dados para a atribuição dos grupos.

Na detecção de eventos anormais em vídeos a *clusterização* tem se mostrado efetiva conforme apresentado na literatura (AHMED et al., 2018). Muitos dos estudos diferenciam os eventos normais dos anormais baseados na distância entre pontos de *clusters* que excedem um limite estabelecido, enquanto outros utilizam a probabilidade de eventos não vistos, onde se a probabilidade for abaixo de um limite estabelecido então o evento é anômalo (Sodemann; Ross; Borghetti, 2012).

A utilização de trajetórias em *clusters* para discriminação de vídeos anômalos foi uma das primeiras métricas estudadas para tal. Foi apresentada na literatura pela primeira vez em 2005 por (NAFTEL; KHALID, 2005) e até hoje se mostra uma das mais efetivas e exploradas métricas. A Figura 7 apresenta um exemplo de agrupamento de trajetórias.

Figura 7 – Resultado do *Clustering* das trajetórias baseado em GPDFSC.



Fonte: (ATHANESIOUS et al., 2019).

Para a *clusterização*, neste trabalho foi utilizada uma abordagem diferente das tradicionais, onde, ao invés de utilizar um algoritmo não supervisionado padrão utilizamos uma meta-heurística. Meta-heurísticas servem para resolver de forma genérica problemas de otimização, normalmente otimização combinatória (PACHECO, 2015), como é o caso da *clusterização*.

### 3.6 Algoritmo Genético

O algoritmo genético é um método de busca meta-heurística que se inspira no processo de seleção natural de Charles Darwin. Parte do princípio de que existem muitos candidatos em um mesmo ambiente mas apenas os mais aptos sobrevivem e se reproduzem. Assim funciona o algoritmo genético, simulando este processo com 3 etapas: seleção, *crossover* (ou cruzamento) e mutação. Estas etapas ocorrem de forma iterativa, aguardando a satisfação de um critério de parada pré-estabelecido, podendo este ser o número de gerações que se deseja obter. Chamamos de cromossomos ou população de indivíduos

os dados de entrada do algoritmo (PACHECO, 2015). A Figura 8 apresenta o fluxograma genérico do algoritmo genético.

Figura 8 – Fluxograma Algoritmo Genético.



Fonte: (PACHECO, 2015).

Em suma, este método é utilizado para resolver problemas de otimização que nada mais são do que os desafios da definição de pontos extremos de uma função, seja minimização ou maximização. O primeiro passo de um processo de otimização consiste na modelagem matemática. A modelagem do problema de otimização é representada pela função objetivo, também conhecida como *fitness* (PACHECO, 2015). Esta função definida será a que o algoritmo irá otimizar seguindo os passos apresentados na Figura 8.

### 3.7 Arquitetura Flexível

Um fator importante observado neste estudo tem relação com a utilização de arquitetura flexível no que tange seus benefícios e aplicações reais. A arquitetura de um sistema nada mais é do que puramente os seus componentes e como estes interagem e se comunicam entre si. Uma abordagem sistemática e projetada produzirá para um sistema uma arquitetura mais flexível, passível de integrações e manutenções mais simples (RABELO, 2019).

Quando isolamos cada parte de um sistema facilitamos a interação entre componentes atuais e futuros, bem como obtemos a possibilidade da reutilização do código. Poder atualizar ou personalizar apenas as partes necessárias de um sistema, simplifica e otimiza a fase de implantação e testes do sistema (RABELO, 2019). A flexibilidade provida pelos arquivos de configuração, a qual pode incluir a especificação de um caminho de instalação/execução ou outros meta-dados que alteram os parâmetros técnicos ou a aparência de um sistema (RABELO, 2019).

## 4 METODOLOGIA

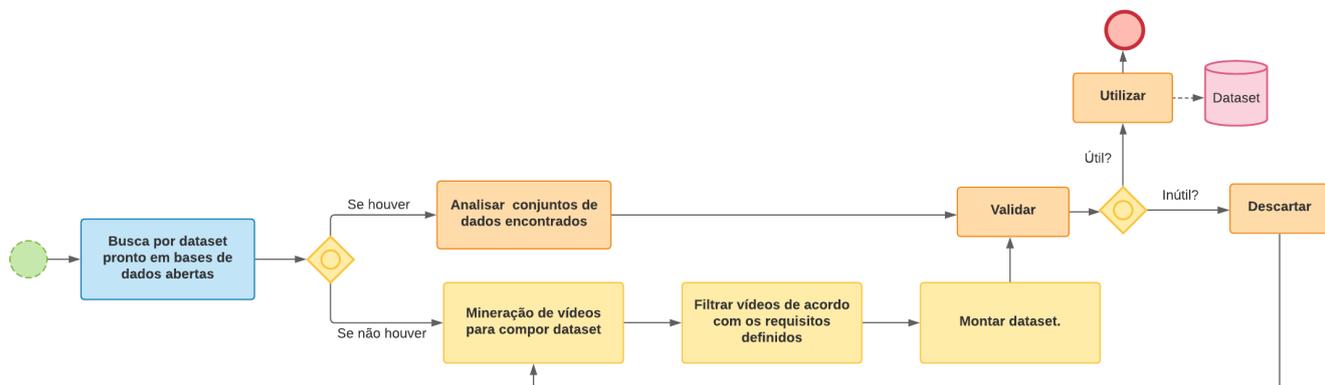
Este capítulo descreve a definição dos processos adotados para a construção deste trabalho, sendo dividido em duas seções. Seção 4.1 referente a definição do *dataset* e seção 4.2 que explica os passos de desenvolvimento da proposta.

### 4.1 Definição do *Dataset*

Neste domínio, para que seja possível propor uma solução, se faz necessário possuir um conjunto de vídeos base que representem algumas situações a serem observadas, nesta seção abordamos de modo geral como foi feita a busca e seleção dos vídeos e na subseção 4.1.1 os critérios da refinação destes resultados estão apresentados.

O processo definido na Figura 9 foi o desempenhado para definir o *dataset* utilizado neste trabalho. A Tabela 4 contém a definição dos requisitos adotados para filtrar os vídeos para compor o *dataset*.

Figura 9 – Processo de Seleção/Criação de *Dataset*.



Fonte: Autor (2021).

Tabela 4 – Critérios para Seleção dos Vídeos.

Perspectiva	Cenário	Objetos
Incluir		
Drone em diferentes níveis de voo	Campo	Somente animais
Câmera de segurança	Galpão	Animais e pessoas
Câmera normal		
Excluir		
Vídeos muito próximos	Ambientes urbanos	Somente pessoas Animais extremamente oclusos

Fonte: Autor (2021).

Como não foram encontrados conjuntos de dados prontos com vídeos de situações normais e possivelmente anormais no campo, foi necessário construir um *dataset* próprio para este trabalho, conforme definido no processo apresentado na Figura 9.

As bases utilizadas foram Videezy Community<sup>1</sup> e Youtube<sup>2</sup>, onde foram selecionados os vídeos que melhor cumpriram os requisitos definidos na Tabela 4. Os dados referentes ao *dataset* construído e palavras chave utilizadas na busca estão apresentados na Tabela 5.

Tabela 5 – *Dataset*.

Base	Quantidade de Vídeos	Palavras Chave
Videezy	59	<i>cow, cattle, gado, vacas, abnormal situation with cattle</i>
Youtube	40	<i>cow, cattle, gado, vacas, abnormal situation with cattle, lida no campo, camperiada</i>

Fonte: Autor (2021).

#### 4.1.1 Definição do Subconjunto de Vídeos

Para delinear melhor o conjunto de vídeos a serem utilizados para o reconhecimento de eventos foi gerado um subconjunto a partir do conjunto de vídeos original obtidos no primeiro passo da busca. No qual os vídeos foram selecionados e recortados de acordo com os momentos de maior interesse. Para esta seleção, foram observados os seguintes critérios:

1. A qualidade de gravação do vídeo, ou seja, ângulo da câmera, estabilidade, foco, iluminação, dentre outros fatores que dificultam o funcionamento do *Yolo* principalmente nos vídeos extraídos do Youtube.
2. A maioria dos vídeos extraídos da plataforma Videezy continham apenas gado, não havendo muito o que analisar se não o comportamento padrão apenas dos animais, o que não é o objetivo deste trabalho.
3. Vídeos muito repetitivos apenas com animais foram descartados pois não agregariam valor considerável na etapa de agrupamento dos dados.

Para essa definição, além destes critérios, a corretude nas detecções e nos rastreamentos de objetos também foram consideradas. Estes dados podem ser encontrados na seção 5.1, onde o subconjunto definido final também é apresentado.

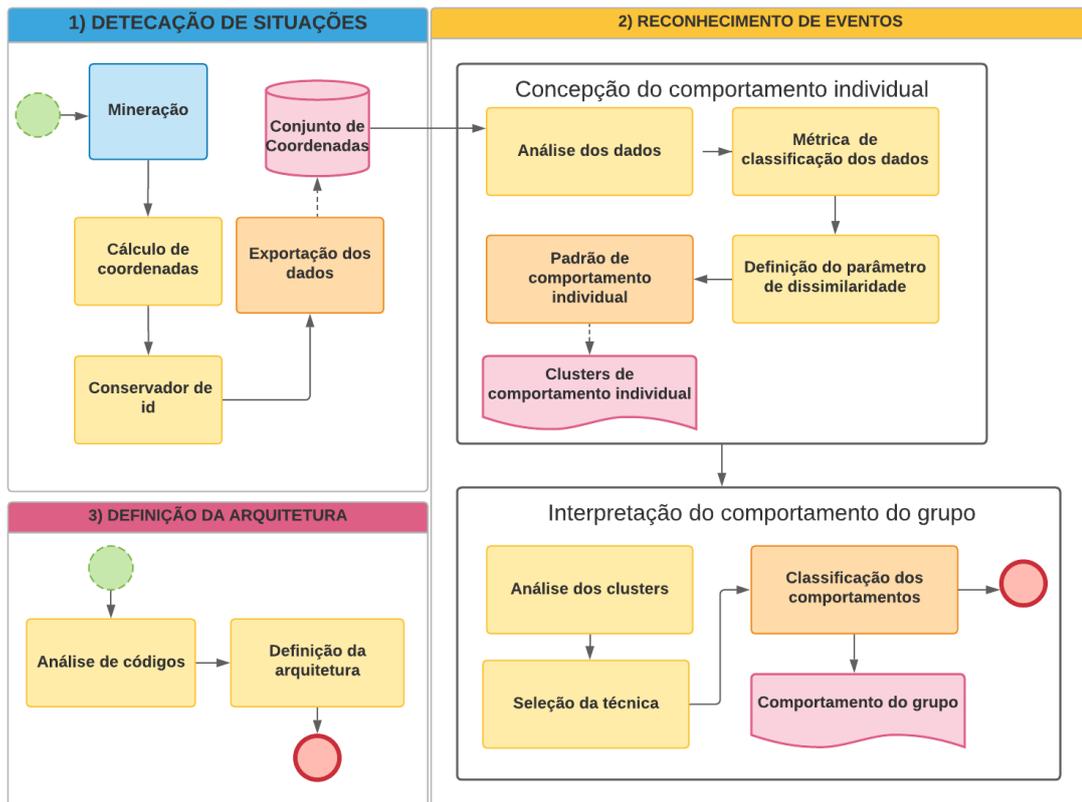
<sup>1</sup> Videezy : <<https://www.videezy.com/free-video/>>

<sup>2</sup> Youtube : <<https://www.youtube.com/>>

## 4.2 Construção da Proposta

O macroprocesso adotado para a construção da proposta deste trabalho pode ser visualizado na Figura 10. O desenvolvimento foi dividido em 3 etapas, sendo estas: Detecção de Situações, Reconhecimento de Eventos e Definição da Arquitetura.

Figura 10 – Macroprocesso Metodológico.



Fonte: Autor (2021).

A etapa de detecção de situações apresentada em 4.2.1 partiu do estudo realizado em Winckler e Thielo (2020) e do algoritmo desenvolvido neste projeto de pesquisa, consistindo predominantemente em um detector de objetos. O reconhecimento de eventos (4.2.2) é a extensão deste mesmo estudo e a ideia de introduzir uma arquitetura flexível (4.2.3) neste contexto tem como objetivo expor que, mesmo em etapas puramente investigativas, é possível introduzir princípios que possuam maior relação com a indústria de software.

Cada um dos passos apresentados na Figura 10 possui definições e outros passos específicos em cada etapa. Na seção 4.2.1 é apresentado o desenvolvimento do algoritmo de detecção de objetos, na 4.2.2 o de reconhecimento de eventos e no 4.2.3 a definição de arquitetura final utilizada.

### 4.2.1 Detecção de Objetos

O objetivo principal deste algoritmo é utilizar a detecção de objetos de modo a extrair as coordenadas de trajetórias e classes dos objetos. Para a definição deste, primeiramente foram minerados repositórios no github para encontrar propostas prontas, dentre os quais o repositório intitulado “gait-recognition-master”<sup>3</sup> foi selecionado, mas infelizmente foi posteriormente retirado do ar.

A linguagem de programação utilizada foi *Python*. É importante ressaltar que os pesos utilizados no algoritmo deste trabalho foram os disponibilizados no repositório “gait-recognition-master”, assim como a estratégia na utilização do *OpenCV*<sup>4</sup> combinadas às detecções do *Yolo*. Os pesos utilizados são de um *Yolo* otimizado no reconhecimento de pessoas. A proposta dos autores foi utilizar uma abordagem no pré-processamento direcionada a retrainar a rede *Yolo* e assim obter uma melhor acurácia na classificação de pessoas. Assumimos que o modelo minerado do repositório é suficiente para a nossa proposta.

Na etapa de refatoração do algoritmo também foram utilizados códigos do repositório “TensorFlow-2.x-*Yolov3*”<sup>5</sup>, como bibliotecas, a fim de melhorar a qualidade do rastreador.

#### 4.2.1.1 Conservador de Id

Cada objeto rastreado possui um código de identificação único, ao qual iremos nos referir simplesmente como Id. Conservadores de Id são utilizados para que seja possível rastrear um mesmo objeto ao longo do vídeo, de modo que seu identificador seja mantido na maior taxa de tempo possível, a fim de prover confiabilidade às trajetórias que serão analisadas, evitando roubos e perdas de Id na medida do possível.

Na busca por conservadores de Id, alguns métodos foram identificados, sendo estes: Mean Shift, Deep Sort, LTSM (Long short-term memory), Filtro de Kalman, Centroid Tracking Distance, os rastreadores nativos do OpenCV e a utilização da referência do *frame* anterior utilizando distância euclideana (MAIYA, 2019).

De acordo com a busca realizada inicialmente, foi desenvolvido um algoritmo inspirado no conceito apresentado em Rosebrock (2018), que consiste na utilização do ponto central de uma *bouding box* para o cálculo de estimativa de posição e foi utilizada a distância euclideana seguindo a proposta de Murugavel (2019) como medida de proximidade entre os pontos centrais, de modo a conservar o Id das detecções. No entanto, como essa primeira proposta não teve o comportamento esperado, passamos a utilizar, assim como em Barreiros et al. (2021), o *Yolo* combinado com o filtro de Kalman para mapear as trajetórias.

<sup>3</sup> gait-recognition-master: <<https://github.com/17614040741/gait-recognition-master>>

<sup>4</sup> OpenCV: <<https://opencv.org>>

<sup>5</sup> TensorFlow-2.x-*Yolov3*: <<https://github.com/pythonlessons/TensorFlow-2.x-Yolov3>>

A utilização do Filtro de Kalman na predição das trajetórias proporciona a obtenção de melhores resultados na preservação de Id, lidando de uma maneira mais eficaz com oclusões e cruzamentos, conforme apresentado em Soares e Thielo (2020). Utilizamos o algoritmo do Filtro de Kalman proposto em “TensorFlow-2.x- YOLOv3”.

#### 4.2.1.2 Cálculo de Coordenadas

Para construção da série temporal dos vídeos, se faz necessário calcular algumas coordenadas que podem ser obtidas através das detecções realizadas com *Yolo*. Primeiramente precisa-se conhecer os vizinhos de *bouding box*, que são os objetos que se encontram em localizações próximas em uma cena. Como o objetivo aqui é saber a distância entre os objetos da cena, optou-se por utilizar a distância euclidiana, estimada de acordo com a equação 4.1. Para definição dos vizinhos, utilizamos um valor limite que, quando ultrapassado, indica a proximidade entre dois objetos.

$$d(\vec{p}, \vec{q}) = \sqrt{\sum_{i=1}^n (q_i - p_i)^2} \quad (4.1)$$

Sendo  $\vec{p}$  e  $\vec{q}$  os pontos a serem aproximados, ou seja, cada um dos lados das *bouding boxes* e  $p_i$  e  $q_i$  são seus respectivos lados. Neste caso em particular, como podemos ir comparando a distância entre todos os pontos de dois em dois, esta medida é apropriada. Além da distância, é importante conhecermos a velocidade de deslocamento dos objetos, a qual pode ser estimada por meio das equações 4.2.

$$\begin{aligned} vx_i &= (x_i(t + \delta t) - x_i(t)) / \delta t \\ vy_i &= (y_i(t + \delta t) - y_i(t)) / \delta t \\ v &= \sqrt{(vx_i^2 + vy_i^2)} \end{aligned} \quad (4.2)$$

Onde  $t$  é o tempo discreto (frame  $i$ ),  $\delta t = 1$  e  $x_i$  e  $y_i$  são os pontos de deslocamento utilizados para calcular a velocidade do objeto nos eixos  $x$  e  $y$  respectivamente. Para calcular a velocidade é necessário conhecer a última posição assumida pelo objeto, bem como a atual.

#### 4.2.1.3 Extração das Coordenadas/Dados

Foram gerados dois arquivos com as coordenadas das trajetórias individuais de cada objeto *frame a frame*, sendo um com os dados individuais do objeto (classe, id, posição, velocidade instantânea, frame), conforme a Tabela 6, e outro com os dados de interações (distância entre os objetos vizinhos mais próximos na cena e colisões), conforme a Tabela 7.

Tabela 6 – Formato dos Dados de Trajetórias.

Id	Classe	Posição	Velocidade	Frame
$\langle int \rangle$	$\langle string \rangle$	[x,y,w,h]	$\langle float \rangle$	$\langle int \rangle$

Fonte: Autor (2021).

Tabela 7 – Formato dos Dados de Interações.

Vizinhos	Colisões	Frame
$\langle int, int \rangle [distâncias]$	$\langle int, int \rangle [distâncias]$	$\langle int \rangle$

Fonte: Autor (2021).

Estes dados extraídos dos vídeos são os que utilizamos nas etapas de classificação dos eventos e trajetórias. O algoritmo completo da extração de coordenadas, detecção e rastreamento de objetos pode ser visualizado no repositório github<sup>6</sup>.

## 4.2.2 Reconhecimento de Eventos

Dos dados gerados na etapa anterior o que nos interessa analisar são os eventos que ocorrem. Sendo assim, deseja-se analisar o comportamento do grupo em um cenário comum, a fim de definir padrões distintos que possam ser encarados como suspeitos. Para tal, primeiramente é necessário identificar os padrões de forma individual e posteriormente analisá-los como um grupo. Por esta razão, a etapa de reconhecimento de eventos foi dividida em duas, sendo a primeira descrita na subseção 4.2.2.1 onde é apresentado o desenvolvimento do classificador de trajetórias e na subseção 4.2.2.2 onde a implementação do filtro de interações entre os objetos é descrita.

### 4.2.2.1 Análise do Comportamento Individual

A abordagem selecionada para a identificação de padrões comuns de deslocamento e trajetória foi a *clusterização* devido à inexistência de um *ground truth* consolidado para o treinamento de algoritmos supervisionados. Para realizar a divisão dos grupos, decidimos utilizar como descritores das trajetórias a velocidade e as acelerações média e máxima de cada objeto. Seguindo este raciocínio, chamamos de matriz de dissimilaridade ( $D$ ) a matriz cujos componentes são obtidos a partir da métrica discriminativa descrita na equação 4.3 aplicada entre pares de trajetórias, a qual representa as características que utilizaremos para diferenciar as ações e agrupá-las.

Conforme apresentado na seção 3.6, optamos pelo algoritmo genético para abordar este problema combinatório devido às suas características inerentes. Além disto, com a revisão de literatura foi possível observar que os métodos tradicionais de *clustering* contam com algumas limitações. Tendo isso em mente, propomos a utilização de um algoritmo

<sup>6</sup> trajectory-tracker: <[https://github.com/SabrinaWinckler/trajectory\\_tracker](https://github.com/SabrinaWinckler/trajectory_tracker)>

genético também para este fim devido a sua escalabilidade, convergência, caráter na resolução de problemas com grande volume de dados e ausência de algumas limitações que métodos de *clustering* tradicionais impõem. Desta forma, objetivamos que com o emprego deste algoritmo estocástico seja possível:

- Avaliar a modelagem dos eventos anormais a fim de identificar sua adequação para este cenário;
- Agrupar as trajetórias de acordo com a matriz de dissimilaridade definida.

Utilizamos a aceleração e velocidade para a definição dos pontos a serem aproximados na nossa matriz  $D$ . A aceleração é dada pela derivada  $a = dv/dt$  que nada mais é do que a taxa de variação da velocidade em relação ao tempo, podendo ser estimada numericamente a partir das sequências de velocidades extraídas no passo 4.2.1. A pergunta que queremos responder com a utilização disto é a seguinte: Quais atributos a trajetória precisa ter para ser discriminada de modo que o erro seja mínimo? Sendo assim, utilizamos a fórmula da distância de Manhattan (SINWAR; KAUSHIK, 2014) como base para nossa função de dissimilaridade a qual determina os valores de nossa matriz  $D$  cujo o elemento  $i, j$  é definido pela equação 4.3.

$$D_{i,j} = \sqrt{(vmax_i - vmax_j)^2} + \sqrt{(amax_i - amax_j)^2} \quad (4.3)$$

Onde  $vmax_i$  e  $vmax_j$  são as velocidades máximas calculadas a partir de um conjunto instantâneo de *frames* do vídeo, o qual deve ser definido de acordo com os segundos em relação aos *frames* que se deseja obter. Por exemplo, considerando 24 *frames* por segundo, se quisermos utilizar 2 segundos do vídeo como base para a comparação da velocidade máxima definiríamos como 48 o espaço de *frames* para nossa identificação. Os índices  $i, j$  denotam um par de trajetórias na equação e  $amax_i$  e  $amax_j$  são as acelerações máximas de um par  $i, j$  de trajetórias que utilizam como base o mesmo espaço de *frames* utilizado para as velocidades.

A matriz de dissimilaridade é a primeira a ser gerada no algoritmo, sendo populada apenas na inicialização e utilizada como referência no restante da execução do algoritmo. Em nossa implementação do algoritmo genético, a população de indivíduos é inicializada de forma aleatória de acordo com o tamanho do arquivo de entrada. Para seleção dos pais para cruzamento no algoritmo genético, utilizamos o método da roleta e a função *fitness* utilizada neste trabalho foi a mesma utilizada por Thielo (2000) em seu estudo, onde apresenta uma função custo que representa a soma das dissimilaridades médias *intra-cluster* como um problema de otimização combinatória, definida de acordo com a equação 4.4.

$$\varepsilon_k^{pc}(\mathbf{M}) = \sum_{v=1}^K \frac{1}{2p_v N} \sum_{i=1}^N \sum_{j=1}^N M_{i,v} M_{j,v} D_{i,j} \quad (4.4)$$

onde  $p_v = \sum_{k=1}^N \frac{M_{kv}}{N}$

Onde  $k$  é o número de *clusters*,  $p_v$  conta o número de trajetórias por *cluster*,  $\mathbf{M}$  é a matriz de cada indivíduo da população onde verificamos com o produto  $M_{i,v}M_{j,v}$  quando os elementos  $i$  e  $j$  pertencem ambos ao *cluster*  $v$ . Por fim,  $D_{i,j}$  é o elemento na posição  $i, j$  da matriz de dissimilaridade definida inicialmente. Apesar do melhor resultado de *clustering* ser obtido com a menor dissimilaridade média intra-*cluster*, o algoritmo genético busca gerar indivíduos com maior *fitness*. Dessa forma, utilizamos como *fitness* o valor negativado da dissimilaridade total, ou seja,  $-\varepsilon_K^{PC}(M)$ .

Para gerar os arquivos de teste para validação, foi utilizada a biblioteca vPython, utilizando como base a simulação proposta por (TAVARES, 2015). Estima-se que os dados estejam agrupados da seguinte forma após a execução do algoritmo:

1. Mudanças muito bruscas, ou seja, mudanças de aceleração e velocidade acentuadas;
2. Comportamento padrão, dado neste estudo como todo movimento constante ao longo do tempo.

O resultado do agrupamento é concatenado ao arquivo de entrada, atribuindo uma classe a cada dado. O algoritmo genético e simulador desenvolvidos pode ser conferidos no repositório github<sup>7</sup>.

#### 4.2.2.2 Interpretação do Comportamento do Grupo

Após esta aproximação, é necessário analisar a trajetória, ou seja, os pontos percorridos e, em seguida cada respectiva classe e id. Com estes dados formaremos a análise comportamental do grupo na qual as trajetórias serão concatenadas com os dados do objeto na análise, visando observar situações de diferentes tipos de colisão e interação entre os objetos de cena. As categorias de eventos de interesse foram definidas como:

1. (CB) Colisão Brusca, que está definida pelo choque entre dois objetos quando pelo menos um deles faz parte de uma trajetória de mudanças muito bruscas;
2. (AB) Afastamento Brusco, que está definida pelo distanciamento entre dois objetos quando pelo menos um deles faz parte de uma trajetória de mudanças muito bruscas;
3. (P) Parada, definida quando um objeto muda sua velocidade para 0 após uma trajetória de movimentos bruscos;

<sup>7</sup> ga-clustering-trajectories: <[https://github.com/SabrinaWinckler/ga\\_clustering\\_trajectories](https://github.com/SabrinaWinckler/ga_clustering_trajectories)>

4. (I) Influência, definida quando um objeto com trajetória de mudanças muito bruscas colide com o outro modifica seu comportamento de modo que a trajetória deste passe a ser brusca também;
5. (O) Oclusão, definida quando o rastreamento de um objeto é perdido atrás de outro;
6. (R) Situações de risco, definidas quando objetos do tipo animal e pessoa mesmo com trajetórias de comportamento padrão interagem ou se aproximam;
7. (NI) Normal ou Inércia, definida quando um objeto se mantém no comportamento padrão.

Somente com as trajetórias anormais não seremos capazes de reconhecer o evento propriamente dito, pois existem situações onde um objeto que se encontra em repouso pode estar fazendo parte de um comportamento anormal. Esta concepção parte do princípio de que para se definir um evento é necessário compreender quem são os envolvidos. Para tal foi desenvolvido um filtro que percorre os dados dos *clusters* e das interações gerados nos passos anteriores.

O algoritmo do filtro funciona de forma semelhante a uma busca sequencial. Ele verifica a classe de cada objeto no frame em questão, faz uma busca pelo mesmo frame nos dados de interação e, a partir disto, compara sua classificação em relação ao estado atual e o próximo instante  $t$  de todos os que colidiram em algum instante, para desta forma classificar o evento obtendo dados sobre as ocorrências de influências e seus diferentes tipos. Por fim, é definido se ocorreu um evento suspeito ou não.

Assim como nas etapas anteriores este algoritmo gera um novo arquivo a partir do arquivo de entrada, adicionando novas colunas que definem o tipo de interação e se é um evento suspeito ou não. Este algoritmo pode ser encontrado também no github<sup>8</sup>.

Com o objetivo de analisar os resultados do filtro e da clusterização foram elaboradas simulações de trajetórias artificiais a fim de verificar o *ground truth* de cada situação. Para gerar as trajetórias necessárias para este estudo, foram realizadas simulações físicas simplificadas dos movimentos dos objetos para cada uma das categorias definidas nesta seção, e visualizadas através da biblioteca vPython.

Na integração final, a proposta consiste em realizar as classificações de maneira sequencial de modo que a entrada seja um vídeo e a saída os dados das detecções após passar por todas as etapas de análise descritas neste Capítulo. Para avaliar o resultado final deste trabalho os resultados foram divididos em três grupos, sendo estes: Acerto parcial, indicado quando a classificação final falhou em até 50% dos casos; Acerto total, quando a classificação final acertou em mais do que 80% dos casos; E erro, quando seu percentual de acerto estiver abaixo de 50%.

<sup>8</sup> filter-feature: <[https://github.com/SabrinaWinckler/filter\\_feature](https://github.com/SabrinaWinckler/filter_feature)>

### 4.2.3 Definição da Arquitetura

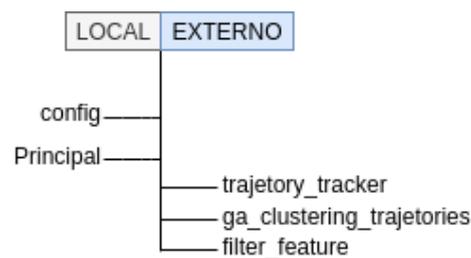
Um dos objetivos deste trabalho é apresentar uma solução para modularização de sistema, de modo a facilitar o desenvolvimento individual de cada módulo e a integração contínua e manutenção, viabilizando assim a produtização.

A proposta consiste em utilizar funções que o git<sup>9</sup> dispõe juntamente com o uso de padrões de projeto adequados para este tipo de problema, analisando os pontos de interesse e necessidades conhecidas.

O primeiro ponto a ser revisado nessa proposta de arquitetura parte igual a todas as outras “Qual o problema que se deseja resolver?”. A partir desta resposta é necessário ter claro em mente todos os pedaços divisíveis do sistema e em quais destes pedaços haverá algum algoritmo de aprendizado de máquina implementado. Mapeando esses já é possível iniciar a especificação da solução que propomos neste trabalho.

Para que as execuções dos algoritmos de aprendizado de máquina não sejam comprometidas e estejam completamente independentes do sistema, foi criado um repositório para cada com os nomes adequados e todas as configurações de um repositório comum. Logo, foi criado um novo repositório, o central, no qual foram definidos os submódulos git que referenciam e puxam o código dos repositórios criados anteriormente. Neste mesmo repositório, foi definida a classe principal e o arquivo de configuração para controle dos submódulos e sequência de execução. A estrutura do repositório central está definida na Figura 11, onde “LOCAL” representa os arquivos criados dentro do repositório e “EXTERNO” os arquivos que vêm de outros repositórios.

Figura 11 – Estrutura do Repositório.



Fonte: Autor (2021).

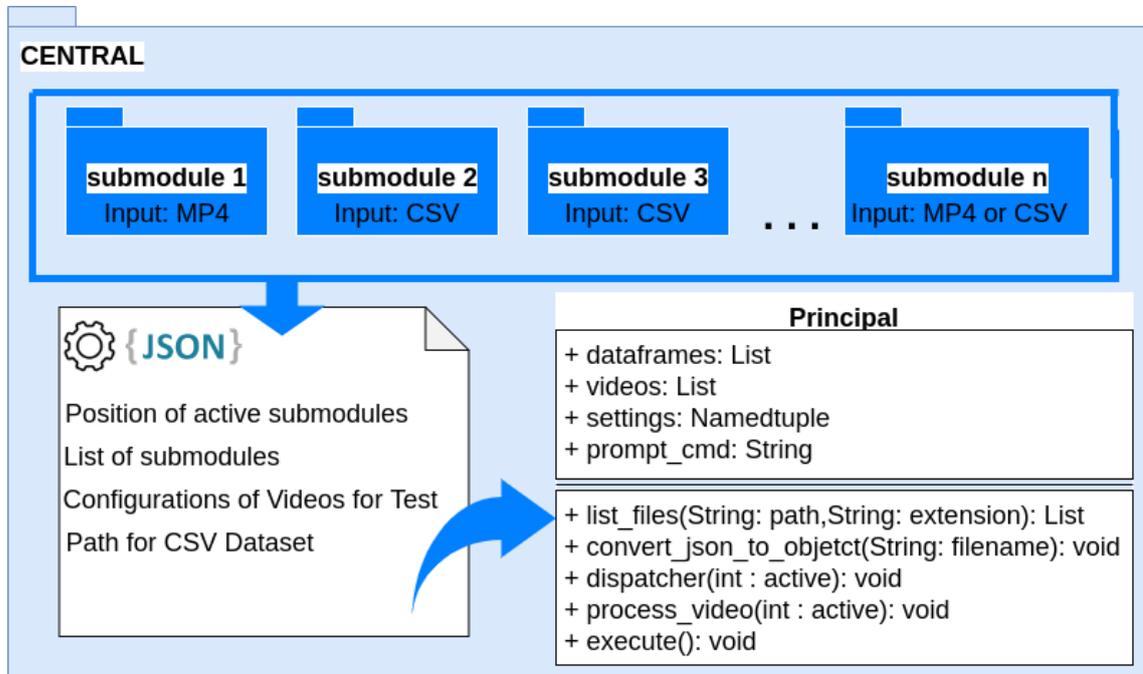
O projeto completo com os submódulos, o arquivo de configuração e a classe principal pode ser conferido no repositório central github<sup>10</sup>. Com a abordagem flexível proposta nesta seção, o sistema poderá ser facilmente modificado, sendo necessário apenas editar o arquivo de configuração respeitando seu padrão e, caso se deseje, incluir/remover ou editar os submódulos do repositório central. A Figura 12 apresenta a generalização da

<sup>9</sup> git: <<https://git-scm.com>>

<sup>10</sup> dsmi:<<https://github.com/SabrinaWinckler/dsmi>>

arquitetura final definida e utilizada no desenvolvimento deste trabalho, a qual contempla todos os pontos abordados nesta seção.

Figura 12 – Arquitetura do Repositório Central.



Fonte: Autor (2021).

O sistema proposto neste estudo possui uma ordem sequencial e entradas esperadas em cada etapa, como pode ser visto na Figura 12. O algoritmo da classe principal foi inspirado na biblioteca *Redux* do *JavaScript* que faz o gerenciamento de estados de aplicativos, executando o sistema de acordo com a configuração do arquivo externo. A Figura 13 apresenta padrão exato do arquivo de configuração no formato *JSON* que é utilizado por esta classe. Este arquivo conta tanto com os dados referentes à execução dos módulos, quanto informações sobre os dados de teste para a fase dos experimentos.

Conforme as definições apresentadas nas Figuras 12 e 13, é possível visualizar a forma como executar e modificar o sistema de modo a este executar o que se deseja neste contexto. Uma vez que, no ambiente corporativo, a integração contínua é um objeto predominante, consideramos que seja de grande valor tanto para a academia quanto para a indústria aproximar a área de pesquisa com o desenvolvimento de produtos. Deste modo, soluções inovadoras antes vistas apenas em artigos científicos tornam-se mais facilmente adaptáveis a novos cenários de aplicação, transformando-se em produtos comercialmente viáveis.

Figura 13 – Arquivo de Configuração.

```
"general" : {
    "active_modules":[0,1,2]
},
"Modules" : [
    {
        "name": "trajetory_tracker",
        "args" : "trajetory_tracker/View.py -v ",
        "entry_type" : ".mp4"
    },
    {
        "name": "ga_clustering_trajetories",
        "args" : "ga_clustering_trajetories/TrajjectoryClassifier.py -f ",
        "entry_type" : ".csv"
    },
    {
        "name": "filter_feature",
        "args" : "filter_feature/Filter.py -c , -i ",
        "entry_type" : ".csv"
    }
],
"Set_test": {
    "abs_path": "trajetory_tracker/test_data/",
    "videos": "Youtube",
    "qnt": 0,
    "just_this": null
},
"Dataset" : {
    "path": "output/trajetory_tracker/test_data/Shortcut/Artificial/"
}
}
```

Fonte: Autor (2021).

## 5 RESULTADOS

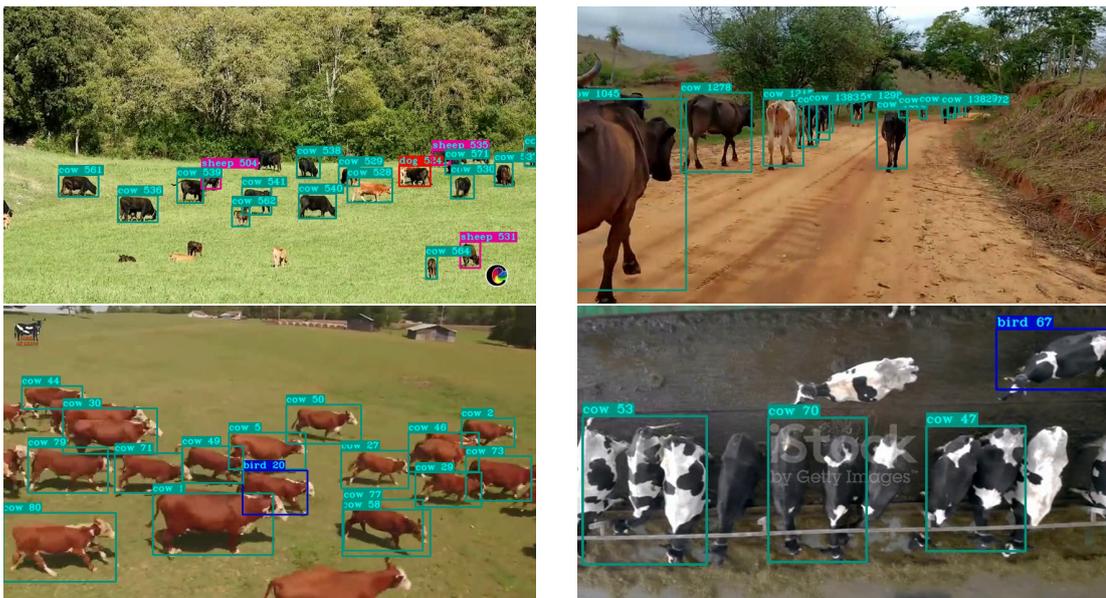
Neste Capítulo estão descritos os resultados obtidos de acordo com os experimentos realizados neste estudo. Os experimentos foram dirigidos de modo a avaliar preliminarmente a qualidade dos resultados de cada um dos algoritmos, bem como a integração entre eles, uma vez que um repositório de dados reais extensivos ou uma métrica consolidada não se encontra disponível. A seção 5.1 descreve os resultados do detector de objetos e a seção 5.2 apresenta os resultados relacionados ao reconhecimento de eventos a partir dos dados obtidos do detector.

### 5.1 Detecção de Objetos

Nesta seção estão apresentados os testes e experimentos realizados com o algoritmo de detecção e rastreamento de objetos. A fim de avaliar a acurácia do modelo *Yolo* em diferentes situações, utilizamos todos os vídeos do *dataset* construído na primeira fase de teste do detector.

A Figura 14 apresenta o funcionamento do detector em alguns vídeos selecionados. Observamos que o detector conta com algumas limitações na detecção de animais aglomerados e distantes da câmera, bem como pode confundir as classes de animais, de modo que essa classificação confusa pode eventualmente comprometer a identificação de cada trajetória específica. No entanto, de modo geral, este poderá não ser um problema tão significativo para o resultado final, uma vez que o que importa no ultimo filtro a ser aplicado é essencialmente a distinção entre animais e pessoas.

Figura 14 – Detecção de Objetos.



Fonte: Autor (2021).

Para avaliar os resultados do Filtro de Kalman comparamos os rastreamentos

com filtro em relação ao rastreamento dos mesmos vídeos utilizando a primeira versão do algoritmo, onde o método para conservar o Id utilizado era baseado na comparação direta das distâncias entre os centros de cada *bounding box*. Após uma análise visual dos rastreamentos em tempo real de cada uma das versões obtivemos uma melhora de aproximadamente 70% com o Filtro de Kalman. A Figura 15 mostra uma situação onde o Filtro de Kalman se mostrou superior ao antigo rastreador.

Figura 15 – Sem Filtro de Kalman x Com Filtro de Kalman.



Fonte: Autor (2021).

Na Figura 15 podemos visualizar uma situação evidenciada, na qual um objeto troca de Id à medida que se move na cena, quando utilizamos a versão sem o filtro de Kalman, já na versão onde utilizamos o filtro de Kalman, essa troca específica não acontece, além de outros rastreamentos ao fundo do vídeo que se mantêm estáveis com o filtro. Desta forma, assim como neste exemplo apresentado, problemas como perda de Id e roubo constantes foram mitigados em grande parte dos vídeos. Apesar da melhora, não foi possível eliminar os erros em sua totalidade. Assim como as oscilações de classes podem

impactar no resultado do nosso classificador de trajetórias, a perda ou roubo contínuo de identificadores tem impacto de mesma proporção.

Tendo isso em vista, a confiabilidade global dos dados foi um aspecto que tivemos que avaliar antes de avançar para o *clustering*. Não conseguimos evitar todas as perdas de id, como já foi mencionado, em parte devido à baixa qualidade do detector e também pela forma como os vídeos do *dataset* coletado foram gravados. Devido a isso foi gerado um subconjunto com as coordenadas dos vídeos com maior corretude no rastreamento dos objetos e com conteúdo relevante para classificar as trajetórias conforme definido na subseção 4.1.1.

Após uma análise visual dos vídeos gerados pelo detector no *dataset* completo, estima-se que a acurácia do modelo *Yolo* utilizado está abaixo de 50% com base nas análises visuais onde comparamos o *ground truth* de cada objeto e classe detectados. Também, de acordo com estas análises as coordenadas dos candidatos à compor o subconjunto foram mapeadas e os vídeos foram recortados de acordo com o evento de interesse, a fim de gerar entradas bem definidas para a etapa de classificação dos dados. Foram selecionados um total de 16 vídeos, sendo 4 do Videezy e 12 do Youtube. A Tabela 8 apresenta a relação de vídeos que compõem o subconjunto gerado. O arquivo de coordenadas de cada um destes vídeos foi utilizado como entrada nos algoritmos de classificação.

## 5.2 Reconhecimento de Eventos

Nesta seção estão descritos os experimentos realizados para a prova de conceito deste trabalho, na qual almeja-se validar a modelagem e identificar os eventos suspeitos. Para tal, dividimos esta seção em duas assim como no desenvolvimento, onde a seção 5.2.1 apresenta a validação das funções de dissimilaridade e testes do algoritmo genético e na seção 5.2.2 estão apresentados os resultados finais do reconhecimento de eventos, com a aplicação do filtro após a aplicação do algoritmo de agrupamento.

### 5.2.1 *Clustering* de Trajetórias

Referente à etapa de análise do comportamento individual apresentada na seção 4.2.2.1 do desenvolvimento, esta descreve os testes realizados e os resultados obtidos. Para validação do funcionamento do algoritmo genético implementado, primeiramente utilizamos números naturais sequenciais de 0 a 10, onde a função de dissimilaridade para aproximar os dados estava definida pela equação 5.1.

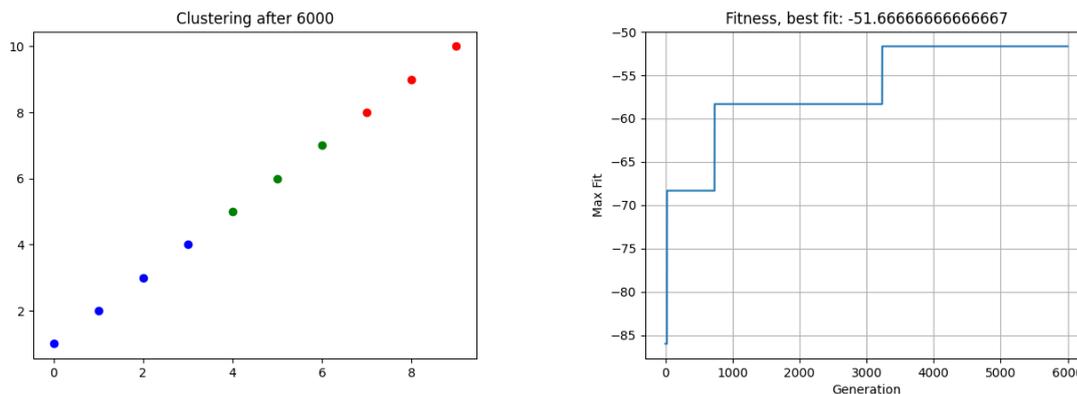
$$D_{i,j} = |x_i - x_j| \quad (5.1)$$

Onde  $x_i$  e  $x_j$  são um par de números naturais sequenciais definidos pelo arquivo de entrada do algoritmo. A Figura 16 apresenta o resultado do *clustering* e a evolução do *fitness* dos dados sequenciais de 0 a 10, o qual conseguiu alcançar sua solução ótima após 6000 gerações.

Tabela 8 – Subconjunto de Dados Reais.

Videezy	
Identificador	Descrição do Vídeo
1	Animais pastando
2	Animais passando por uma pessoa a cavalo
3	Pessoas andando entre animais
4	Pessoas andando entre animais
Youtube	
Identificador	Descrição do Vídeo
5	Pessoa guiando animais
6	Pessoas colocando animal na carreta
7	Pessoas guiando animais a cavalo
8	Pessoa abatendo animal com arma de fogo
9	Pessoas perturbando animal preso
10	Pessoas observando animais ao redor delas
11	Pessoa acariciando animal
12	Pessoa segurando animal à força
13	Pessoa guiando animais
14	Pessoas guiando animais
15	Animal sendo segurado à força por pessoas
16	Animal sendo arreado e puxado por pessoa

Fonte: Autor (2021).

Figura 16 – *Clustering* e *Fitness* de 10 Números Sequenciais.

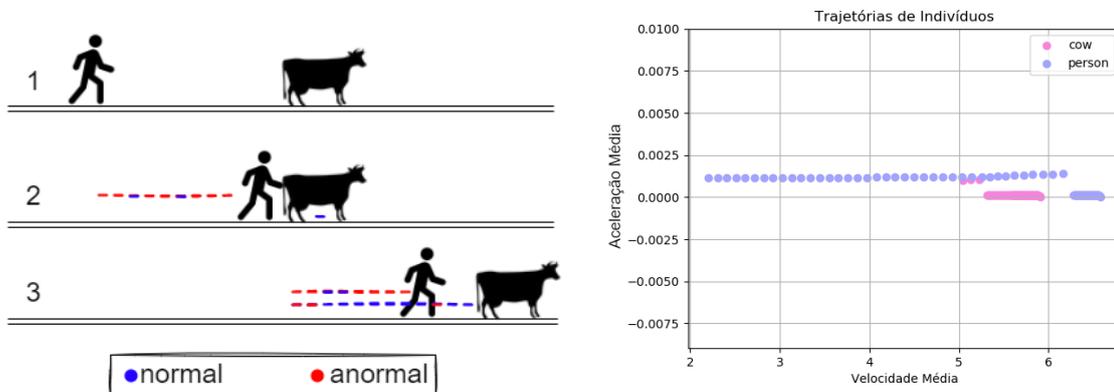
Fonte: Autor (2021).

Na sequência, após validar o funcionamento do algoritmo genético, passamos para a fase de validação da nossa função de dissimilaridade para *clustering*. Por conta das diferenças entre o cenário de interesse selecionado neste estudo e os casos estudados na literatura, não será possível realizar uma comparação direta dos resultados obtidos com os resultados dos trabalhos relacionados. Isso se dá pelo fato de que, em nosso caso, os

comportamentos normais ou anormais observados exclusivamente entre pessoas, que é o caso abordado na literatura, possuem características muito diferentes do comportamento anormal que objetivamos identificar entre pessoas e animais. Sendo assim, utilizamos dados de trajetórias artificiais cuidadosamente elaboradas, conforme definidas na seção 4.2.2.2, a fim de investigar uma função de dissimilaridade capaz de distinguir os dados apropriadamente.

A Figura 17 é a representação visual da simulação de trajetórias artificiais onde acontece a influência de um objeto sobre o outro com interação e proximidades de risco, sendo estes uma pessoa e um animal, e apresenta os valores médios de aceleração e velocidade dos objetos destas mesmas trajetórias artificiais.

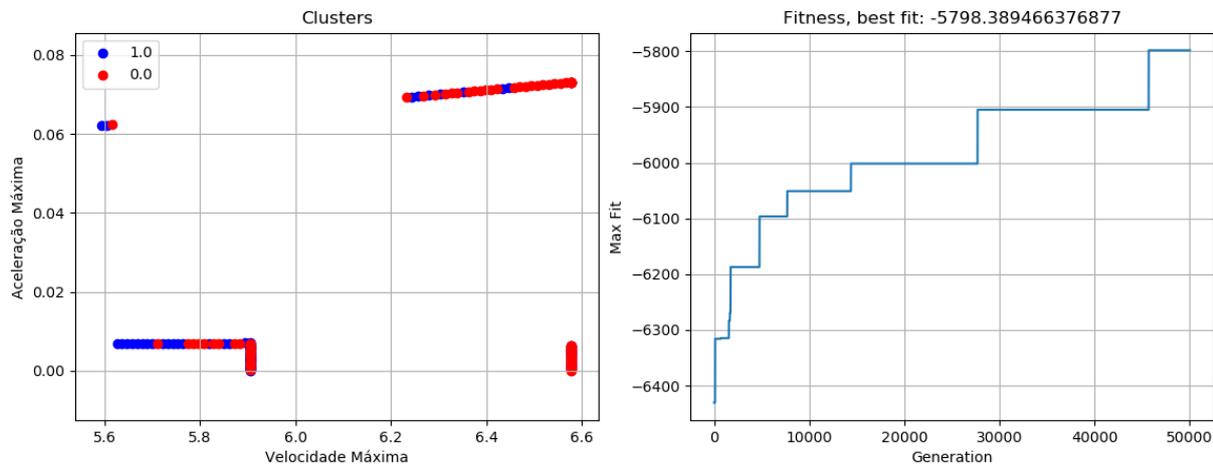
Figura 17 – Representação de Trajetórias Artificiais e os seus Valores Médios.



Fonte: Autor (2021).

Na Figura 17 é possível observar que as acelerações e velocidades dos objetos simulados são constantes na maior parte do tempo e que cada uma das trajetórias artificiais apresentam um esboço de seu agrupamento por segmento. O agrupamento exato destas trajetórias pode ser visualizado na Figura 18 onde mostramos o *fitness* obtido do algoritmo genético após 50 mil gerações para estas trajetórias artificiais e o *clustering* baseado na aceleração e velocidade máximas dos segmentos de trajetórias compostos por janelas de 61 *frames* cada, sendo que no eixo horizontal se encontram as velocidades máximas e no eixo vertical a acelerações máximas. No gráfico do *fitness* podemos ver sua evolução de valores ao longo das 50 mil gerações, o qual se estabiliza após 30 mil gerações e tem um novo salto de evolução próximo a 50 mil gerações.

O gráfico de *clustering* apresentado na Figura 18 condensa informações sobre os segmentos de duas trajetórias de modo que possamos avaliar a discriminação simultânea de todos os objetos na cena. Para analisar individualmente o agrupamento de cada trajetória, podemos observar a Figura 19, a qual apresenta o resultado do agrupamento dos segmentos de cada uma das trajetórias individuais, lembrando que estas são baseadas nas simulações previamente descritas na Figura 17, onde os segmentos das trajetórias

Figura 18 – *Clustering e Fitness* das Trajetórias Artificiais.

Fonte: Autor (2021).

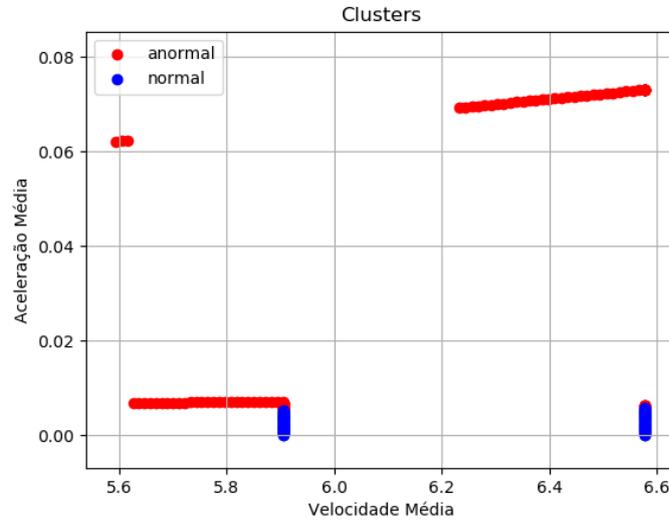
estão coloridos de modo a representar um esboço de seu agrupamento.

Ao avaliar os gráficos dispostos na Figura 19, podemos delinear o comportamento individual de ambos os objetos, sendo que cada um possui valores bem distintos de aceleração e velocidade no início do movimento e, em um certo momento em função interação, os dois objetos apresentam o mesmo valor de aceleração máxima. Além disto, outro fator observado é que enquanto um objeto parte de um comportamento normal e avança para um anormal, outro objeto parte de um comportamento anormal tendo momentos em que este se normaliza mas em sua totalidade a trajetória é definida como anormal.

Para avaliar o resultado obtido, realizamos a ordenação dos dados em função dos seus valores de velocidade, e dividimos o *dataset* ao meio similarmente ao que foi feito no primeiro teste com os números. Em seguida, o algoritmo foi executado com apenas uma geração para observar o valor do *fitness* que corresponderia à melhor configuração de agrupamento. A Figura 20 apresenta os possíveis *clusters*, baseados nos valores máximos de velocidade e aceleração de grupos de *frames* esperado, sendo que o valor do *fitness* obtido após uma geração foi igual a  $-6185,292733411756$ .

Apesar da tentativa de inferir o resultado ótimo manualmente através da ordenação prévia do *dataset*, esta não correspondeu à melhor solução possível uma vez que, quando comparamos o seu *fitness* com o obtido após a execução real do algoritmo na Figura 18, observamos que a execução sem manipulação dos dados foi capaz de obter um maior valor de *fitness* e um agrupamento melhor de um modo geral (avaliado visualmente). Sendo assim, o resultado do agrupamento foi considerado satisfatório, e uma vez que os dados das trajetórias artificiais se assemelham aos dados reais, assumimos a viabilidade de utilizar esta função de dissimilaridade para a separação dos dados em *clusters*. No entanto, antes de avançar para esta etapa, foram testadas algumas funções de dissimilaridade alternativas, a fim de verificar se a função utilizada até então é a mais adequada para este



Figura 20 – Possíveis *Clusters* dos Dados das Trajetórias Artificiais Esperados.

Fonte: Autor (2021).

5.2.

$$D_{i,j} = \frac{vmax_i^i + vmax_j^j}{2} + \frac{amax_i^i + amax_j^j}{2} \quad (5.2)$$

Onde  $D$  é a própria matriz,  $vmax_i$  e  $vmax_j$  são as velocidades máximas de um par  $i, j$  de trajetórias e  $amax_i$  e  $amax_j$  são as acelerações máximas de um par  $i, j$  de trajetórias e  $i$  e  $j$  representam a posição instantânea da trajetória no conjunto de dados.

A função 5.2 obteve um valor de *fitness* maior do que o da função 4.3 e agrupou os dados de maneira semelhante. O gráfico dos *clusters* apresentado na Tabela 9 é baseado nos valores médios de aceleração e velocidade calculados de acordo com janelas de 61 *frames* assumindo uma taxa de 24 fps. Ambos os agrupamentos estão dispostos de modo que um dos grupos se encontra mais próximo do eixo horizontal e outro mais disperso, sendo que a função 5.2 apresentou o melhor desempenho na discriminação dos elementos. Por conta da semelhança entre os resultados obtidos entre a distância baseada em polinômios de Chebyshev e a distância de Manhattan, a função de dissimilaridade adotada permaneceu sendo a de Manhattan. Vale ressaltar que estes são resultados preliminares e apenas estas características provavelmente não serão suficientes para descrever todos os casos de interesse nas trajetórias em uma situação real.

### 5.2.2 Filtro de Interação

Esta seção apresenta os resultados referentes à seção 4.2.2.2 do desenvolvimento que tem como objetivo a interpretação do comportamento do grupo. Diferente da primeira versão do algoritmo, onde na detecção de objetos já fazíamos considerações, como pode ser visto na Figura 21, não consideramos mais apenas a proximidade entre os elementos

Tabela 9 – Teste de Funções de Dissimilaridade.

Função $D_{i,j}$	Fitness	Clustering
5.2	-5.735250713319676e+86	
4.3	-16539.699216367648	

Fonte: Autor (2021).

como eventos de risco. Agora iremos, na etapa de pós processamento, classificar o que cada aproximação e colisão representam utilizando as coordenadas extraídas do vídeo juntamente com o *cluster* que o algoritmo genético atribuiu a cada dado.

Figura 21 – Identificação de Suspeitos com Base na Distância.



Fonte: Autor (2021).

Ainda assumindo a premissa de que os donos dos animais e os tratadores não se encontram junto a estes animais, os dados receberam rótulos para suas interações. O resultado deste algoritmo para as 6 trajetórias artificiais primordiais pode ser visualizado na Tabela 10, onde o padrão que cada trajetória artificial representa está disposto em Figuras representativas de acordo com a simulação gerada e os resultados obtidos descritos de acordo com a definição descrita em 4.2.2.2. As trajetórias artificiais primordiais são todas aquelas que buscam descrever de um tipo de interação específica, podendo ainda possuir outros eventos em sua constituição.

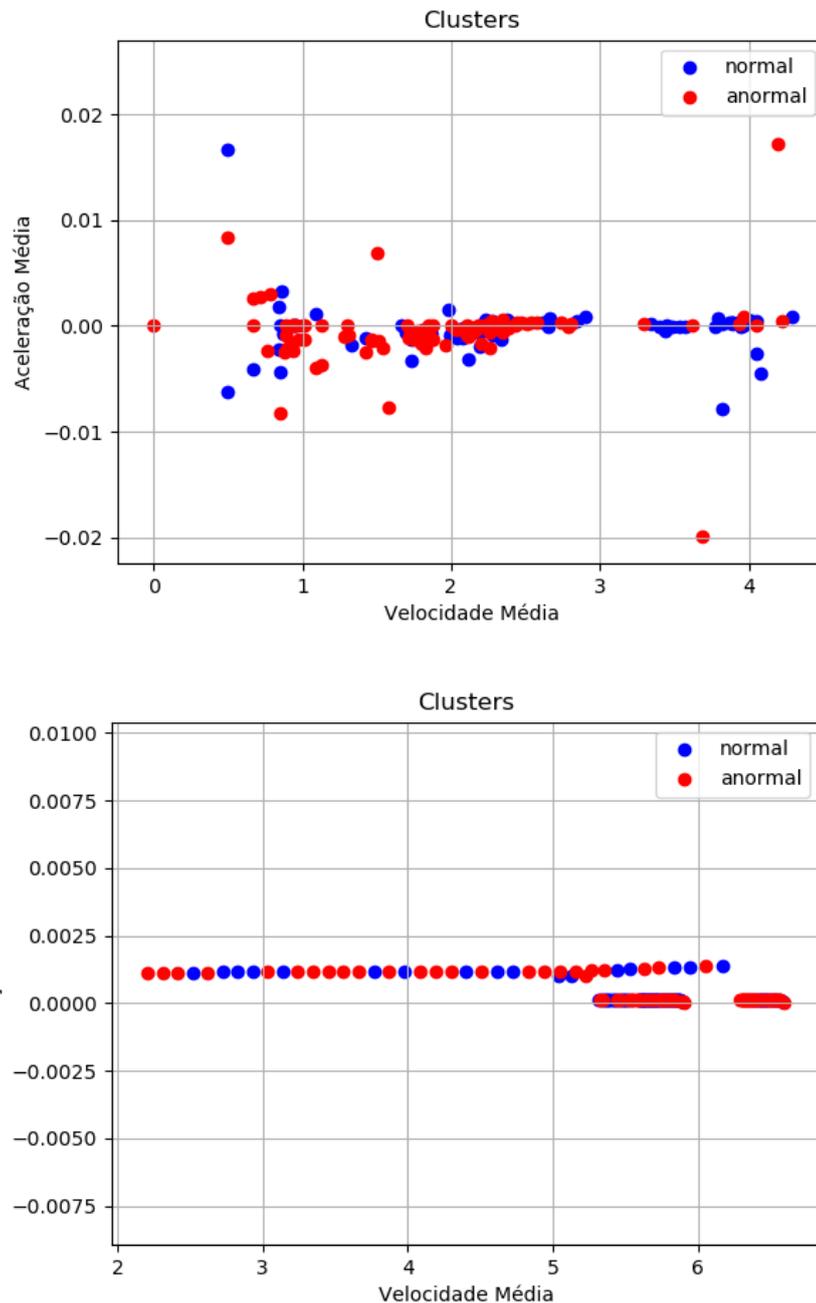
De acordo com os dados de saída do filtro apresentados na Tabela 10, podemos verificar que os agrupamentos das trajetórias com o algoritmo genético forneceram entradas apropriadas para o filtro, o qual rotulou de maneira satisfatória as trajetórias onde, em todos os casos, o resultado obtido foi o esperado. Também foram geradas combinações das trajetórias primordiais, das quais resultaram 127 diferentes situações de trajetórias possíveis para teste do algoritmo mas, devido ao alto custo computacional envolvido pelo grande volume de dados não foi possível utilizá-las. Por outro lado, assumindo os resultados de cada classe individual, de acordo com o apresentado na Tabela 10, estima-se que as diferentes combinações de eventos teriam o mesmo desempenho positivo. Sendo assim, assumindo que o filtro e o algoritmo genético possuem as características necessárias para processar os dados reais, iremos, da mesma forma que apresentamos os resultados das trajetórias artificiais, apresentar os resultados para os dados reais. Como na maioria dos dados artificiais após 30 mil gerações o *fitness* se estabilizava, conforme pode ser visto na Figura 18, os dados reais menores foram processados com 30 mil gerações mas, devido ao custo computacional significativamente mais elevado, os dados mais extensos precisaram ser executados com um número menor de gerações (10 mil). A Figura 22 apresenta o resultado do *clustering* baseado nos valores médios da aceleração e velocidade do vídeo 5 dos dados reais, conforme numeração na Tabela 8, em comparação com os resultados para os dados artificiais que representam um afastamento brusco entre dois objetos. Estes dois conjuntos de trajetórias foram comparados já que no vídeo real temos algumas situações de afastamento brusco assim como na artificial.

Ao observarmos esta representação é possível perceber que a *clusterização* com os dados reais foi semelhante à *clusterização* com os dados artificiais. Sendo assim, verificamos a hipótese referente ao comportamento do algoritmo para os dados de trajetórias reais. Outro fator que podemos perceber tem relação com a diferença nas variações de aceleração dos dados reais em comparação com os artificiais, os quais apresentam menor regularidade.

Em seguida selecionamos os vídeos mais distintos do subconjunto obtido a partir dos dados reais para demonstrar seu resultado gerando gráficos que representam de forma global os padrões identificados no decorrer dos *frames* e sua classificação instantânea, nos quais a classificação de evento suspeito é representada por 0 (Falso) ou 1 (Verdadeiro).

Tabela 10 – Resultados com Trajetórias Artificiais.

Padrão	Esperado	Obtido	Suspeito
	Colisão Brusca	Colisão Brusca	Sim
	Parada	Parada	Sim
	Oclusão	Oclusão	Sim
	Afastamento Brusco	Afastamento Brusco	Sim
	Influência	Influência	Sim
	Normal ou Inércia	Normal ou Inércia	Não
	Situação de Risco	Situação de Risco	Sim

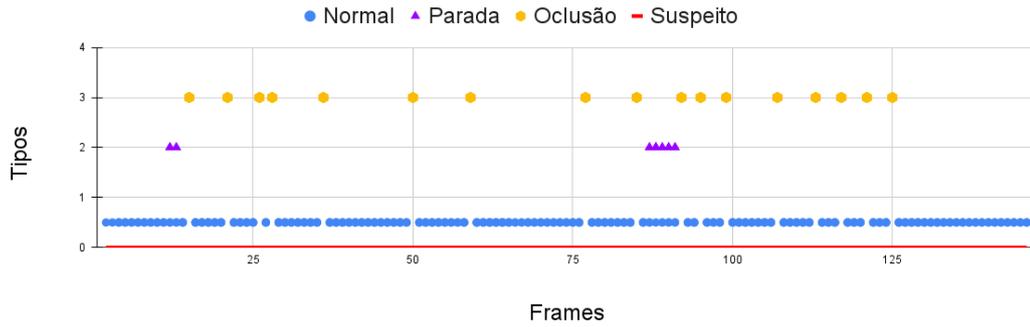
Figura 22 – *Clustering* de uma Trajetória Real vs Artificial.

Fonte: Autor (2021).

A Figura 23 apresenta o gráfico da saída do filtro com os dados do vídeo 1, conforme sua numeração na Tabela 8. É possível verificar neste gráfico que, apesar das oclusões e paradas, na maior parte do tempo as interações entre os objetos foram classificadas como normais assim como em nenhum instante suas trajetórias foram consideradas suspeitas, características estas que nosso filtro consegue identificar satisfatoriamente.

Já a Figura 24 apresenta o gráfico da saída do filtro com os dados do vídeo 5, conforme sua numeração na Tabela 8. As interações identificadas foram condizentes

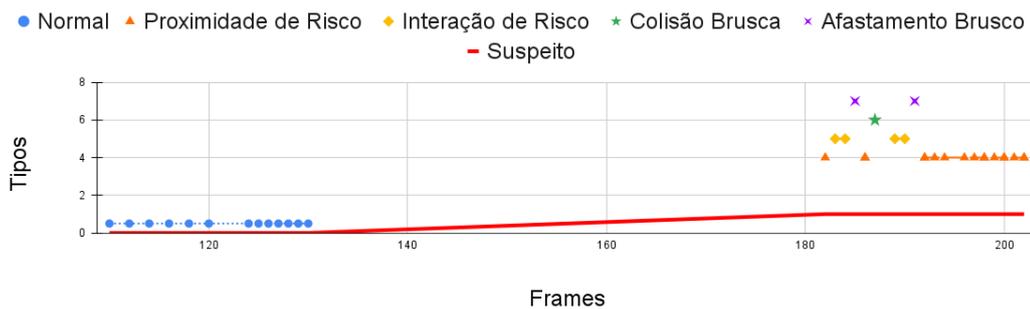
Figura 23 – Gráfico da Trajetória Real 1.



Fonte: Autor (2021).

com o esperado neste vídeo, sendo estas: proximidade de risco, interação de risco, colisão brusca e afastamento brusco enquanto a pessoa guiava os animais até seu local de destino.

Figura 24 – Gráfico da Trajetória Real 5.



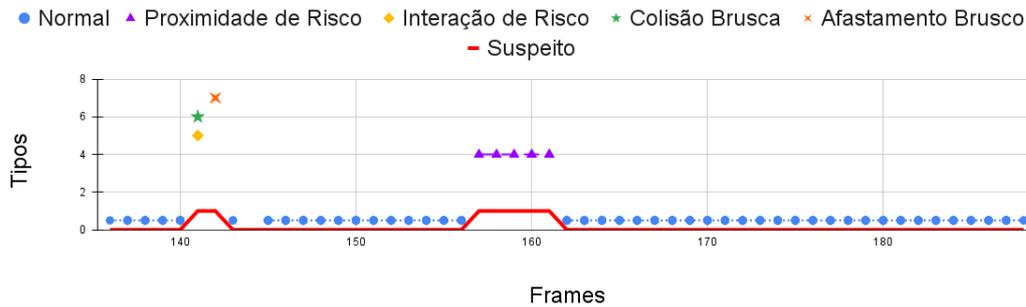
Fonte: Autor (2021).

Na Figura 25 podemos visualizar o gráfico da saída do filtro com os dados do vídeo 3, conforme sua numeração na Tabela 8. Neste caso, vemos vários momentos de normalidade dentre os eventos suspeitos ocorridos enquanto as pessoas andavam entre os animais que foram, também, coerentes com o que pudemos perceber visualmente no vídeo com as detecções. Este aspecto também verifica a variedade de situações/combinções que o filtro consegue identificar.

Por sua vez, a Figura 26 apresenta o gráfico da saída do filtro com os dados do vídeo 8, conforme sua numeração na Tabela 8. Neste caso, obtemos um resultado satisfatório, considerando que o animal foi abatido com arma de fogo e o mais anormal que ocorreu entre os objetos foi, de fato, a proximidade de risco.

A Figura 27 apresenta o gráfico da saída do filtro com os dados do vídeo 9, conforme sua numeração na Tabela 8. Neste caso, podemos visualizar uma situação um tanto diferente, onde não necessariamente em todos os momentos em que alguma interação suspeita aconteceu estava ocorrendo uma trajetória suspeita. Desta forma, comprovamos

Figura 25 – Gráfico da Trajetória Real 3.



Fonte: Autor (2021).

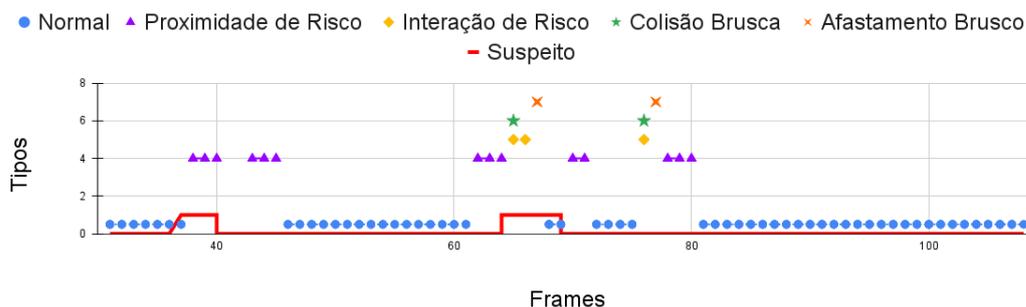
Figura 26 – Gráfico da Trajetória Real 8.



Fonte: Autor (2021).

a eficácia da combinação dos métodos propostos neste estudo, onde utilizamos não só dados isolados dos objetos mas também dados de interações entre estes objetos.

Figura 27 – Gráfico da Trajetória Real 9.



Fonte: Autor (2021).

A Figura 28 apresenta o gráfico da saída do filtro com os dados do vídeo 11, conforme sua numeração na Tabela 8. Aqui, verificamos também um caso distinto, onde na maior parte do tempo do vídeo uma trajetória anormal está acontecendo e em nenhum momento a interação é identificada como normal devido à constante proximidade e contato entre os objetos da cena.

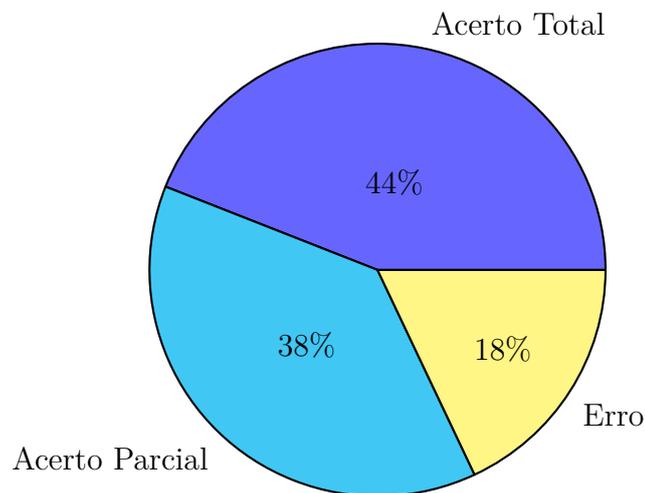
Figura 28 – Gráfico da Trajetória Real 11.



Fonte: Autor (2021).

Conforme o observado, podem haver mudanças bruscas de velocidade e ainda assim este não ser um evento suspeito, assim como podem haver momentos de normalidade dentre os eventos suspeitos. Os resultados demonstram que o algoritmo proposto consegue inclusive identificar esses casos com distintas variações e combinações. O gráfico na Figura 29 apresenta o percentual de acerto do filtro com todos os vídeos reais do subconjunto, sendo que para esta avaliação foi utilizado um critério qualitativo, no qual o vídeo foi analisado em comparação com o gráfico de cada trajetória para definir o percentual que ele acertou considerando a classificação indicativa por *frame*.

Figura 29 – Resultados do Subconjunto com os Vídeos Reais.



Fonte: Autor (2021).

Quando comparamos os dados artificiais com os reais, é possível observar que obtemos resultados positivos em ambos, apesar dos erros em alguns casos e da sensibilidade que o detector de movimentos que sofre pelas influências dos ruídos das *bounding boxes*. Os resultados obtidos com os dados reais foram bastante satisfatórios apesar dos percentuais de erro de 18% e de acerto parcial de 38%. Devemos considerar que este trabalho é apenas o início de uma investigação de detecção de eventos anormais em zona rural,

sendo necessários estudos mais aprofundados, a fim de encontrar e delinear os eventos de melhor forma. Estimamos que, com mais refatorações, para as quais nossos algoritmos já estão preparados, devido à sua arquitetura de caráter flexível, este sistema tem potencial para produzir melhores resultados futuramente.

## 6 CONSIDERAÇÕES FINAIS

Neste trabalho foi possível demonstrar a importância do estudo e aprofundamento de técnicas para o cenário em questão devido à problemática iminente dos crimes na zona rural. Observou-se, também, que para resultados mais exatos precisaríamos de um *dataset* do domínio escolhido no qual o estado da câmera fosse conhecido e constante. Ou seja, onde a velocidade e posição do drone/câmera fosse conhecida de modo que fosse possível normalizar os dados e não gerar falsas velocidades/acelerações, bem como o ângulo da câmera fosse ajustado de modo a não haver falsas detecções de colisões. Como o *subset* definido neste trabalho é híbrido, ou seja, não possui apenas filmagens de drones e a velocidade de movimentação das câmeras é desconhecida em todos os vídeos, este fator foi ignorado a fim de conseguir alcançar os objetivos definidos neste trabalho.

Além disto, no que tange ao desempenho computacional do algoritmo genético foi possível observar as limitações de processamento em relação ao tempo, pois este exigia muitas gerações de acordo com o número de dados que desejássemos agrupar, sendo necessários dias para a visualização do agrupamento de um vídeo de 5 minutos.

Outro fator importante observado no desenvolvimento deste trabalho diz respeito às práticas de engenharia de software utilizadas neste trabalho. Esta aplicação auxiliou de maneira significativa nos testes de diferentes funções, manutenção, integração e automatização de tarefas para processamento dos dados desde a entrada do detector até a saída do filtro.

Ainda existe um longo caminho a ser trilhado para que possamos, com auxílio da tecnologia, realmente prestar socorro aos produtores rurais. Como trabalhos futuros, propomos o retreinamento do modelo *Yolo* para melhorar a confiabilidade dos dados, implementação de um filtro para redução de ruído das *bounding boxes*, um aprofundamento na análise dos tipos de relações e correlações entre objetos para gerar mais classes de saída do filtro de características e a construção de um *dataset* específico para eventos suspeitos em ambiente rural. Uma outra possibilidade, que estamos investigando atualmente, diz respeito à utilização de vídeos criados a partir do jogo *Red Dead Redemption* como entrada para o detector de objetos, a fim de gerar dados de trajetórias com o controle da câmera do jogo e o estudo desenvolvimento de um sistema de alertas em tempo real que faça comunicação com entidades de segurança, de modo a estas receberem as imagens de que causaram o alerta bem como a geolocalização da propriedade na qual ocorreu.



## REFERÊNCIAS

- AHMED, S. A. et al. Trajectory-based surveillance analysis: A survey. **IEEE Transactions on Circuits and Systems for Video Technology**, IEEE, v. 29, n. 7, p. 1985–1997, 2018. Citado 3 vezes nas páginas 35, 36 e 37.
- AKÖZ, O.; KARSLIGIL, M. E. Video-based traffic accident analysis at intersections using partial vehicle trajectories. In: **CIVR 2010 - 2010 ACM International Conference on Image and Video Retrieval**. Diyarbakir, Turkey: IEEE, 2010. p. 335–341. Citado na página 27.
- AKÖZ, ; KARSLIGIL, M. E. Severity detection of traffic accidents at intersections based on vehicle motion analysis and multiphase linear regression. In: **IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC**. Funchal, Portugal: IEEE, 2010. p. 474–479. Citado na página 26.
- ALVES, G. **Detecção de Objetos com YOLO – Uma abordagem moderna**. 2020. [Online; accessed 20-july-2021]. Disponível em: <<https://iaexpert.academy/2020/10/13/deteccao-de-objetos-com-yolo-uma-abordagem-moderna/>>. Citado na página 34.
- ANJUM, N.; CAVALLARO, A. Multifeature object trajectory clustering for video analysis. **IEEE Transactions on Circuits and Systems for Video Technology**, IEEE, v. 18, n. 11, p. 1555–1564, 2008. Citado na página 26.
- ATHANESIOUS, J. et al. Detecting abnormal events in traffic video surveillance using superior orientation optical flow feature. **IET Image Processing**, v. 14, n. 9, p. 1813–1821, 2020. Citado na página 26.
- ATHANESIOUS, J. J. et al. Trajectory based abnormal event detection in video traffic surveillance using general potential data field with spectral clustering. **Multimedia Tools and Applications**, v. 78, n. 14, p. 19877–19903, 2019. Citado 6 vezes nas páginas 26, 27, 29, 30, 36 e 37.
- ATHANESIOUS, J. J. et al. Adaptive density based data mining technique for detection of abnormalities in traffic video surveillance. **Journal of Intelligent and Fuzzy Systems**, v. 39, n. 3, p. 3737–3747, 2020. Citado na página 30.
- BARN, B.; BARAT, S.; CLARK, T. Conducting systematic literature reviews and systematic mapping studies. In: **Proceedings of the 10th Innovations in Software Engineering Conference**. New York, NY, USA: Association for Computing Machinery, 2017. (ISEC '17), p. 212–213. ISBN 9781450348560. Disponível em: <<https://doi.org/10.1145/3021460.3021489>>. Citado na página 19.
- BARREIROS, M. de O. et al. Zebrafish tracking using yolov2 and kalman filter. **Scientific Reports**, Nature Publishing Group, v. 11, n. 1, p. 1–14, 2021. Citado na página 42.
- BEYAN, C.; FISHER, R. B. Detecting abnormal fish trajectories using clustered and labeled data. In: **2013 IEEE International Conference on Image Processing, ICIP 2013 - Proceedings**. Melbourne, VIC, Australia: IEEE, 2013. p. 1476–1480. Citado 2 vezes nas páginas 26 e 27.
- BOYD, J. P. Chebyshev and fourier spectral methods. Courier Corporation, 2001. Citado 2 vezes nas páginas 31 e 57.

BRUN, L.; SAGGESE, A.; VENTO, M. Learning and classification of car trajectories in road video by string kernels. In: **VISAPP 2013 - Proceedings of the International Conference on Computer Vision Theory and Applications**. Barcelone, Spain.: hal.archives-ouvertes.fr, 2013. v. 1, p. 709–714. Citado na página 26.

CHEN, D. .; HUANG, P. . Dynamic human crowd modeling and its application to anomalous events detection. In: **2010 IEEE International Conference on Multimedia and Expo, ICME 2010**. Singapore: IEEE, 2010. p. 1582–1587. Citado na página 26.

CHEN, D. .; HUANG, P. . Motion-based unusual event detection in human crowds. **Journal of Visual Communication and Image Representation**, v. 22, n. 2, p. 178–186, 2011. Citado na página 26.

CNA, A. de C. **Instituto CNA participa de debate sobre furto de gado**. 2021. [Online; accessed 10-july-2021]. Disponível em: <<https://www.cnabrasil.org.br/noticias/instituto-cna-participa-de-debate-sobre-furto-de-gado>>. Citado 2 vezes nas páginas 16 e 17.

CUI, J.; LIU, W.; XING, W. W. Crowd behaviors analysis and abnormal detection based on surveillance data. In: **Proceedings: DMS 2014 - 20th International Conference on Distributed Multimedia Systems**. [S.l.]: Elsevier, 2014. p. 3–7. Citado na página 26.

EDDY, S. R. Hidden markov models. **Current opinion in structural biology**, Elsevier, v. 6, n. 3, p. 361–365, 1996. Citado na página 29.

ESPINOSA-ISIDRÓN, D. L.; GARCÍA-REYES, E. B. **A new dissimilarity measure for trajectories with applications in anomaly detection**. Berlin, Heidelberg: Springer, 2010. v. 6419 LNCS. 193-201 p. (Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), v. 6419 LNCS). Citado 2 vezes nas páginas 26 e 31.

EUSGELD, I.; FREILING, F.; REUSSNER, R. H. **Dependability Metrics: GI-Dagstuhl Research Seminar, Dagstuhl Castle, Germany, October 5-November 1, 2005, Advanced Lectures**. Verlag Berlin Heidelberg: Springer, 2008. v. 4909. Citado na página 21.

FEIZI, A. Hierarchical detection of abnormal behaviors in video surveillance through modeling normal behaviors based on auc maximization. **Soft Computing**, v. 24, n. 14, p. 10401–10413, 2020. Citado na página 26.

FEIZI, A.; AGHAGOLZADEH, A.; SEYEDARABI, H. Application of combined local object based features and cluster fusion for the behaviors recognition and detection of abnormal behaviors. **International Journal of Engineering, Transactions B: Applications**, v. 28, n. 11, p. 1597–1604, 2015. Citado 2 vezes nas páginas 26 e 27.

FENDRI, E.; GHRAB, N. B.; HAMMAMI, M. Clustering based on two layers for abnormal event detection in video surveillance. **International Journal of Software Innovation**, v. 5, n. 4, p. 1–18, 2017. Citado na página 27.

FERNANDO, W. S. K. et al. Video event classification and anomaly identification using spectral clustering. In: **15th International Conference on Advances in ICT for Emerging Regions, ICTer 2015 - Conference Proceedings**. Colombo, Sri Lanka: IEEE, 2016. p. 21–24. Citado na página 26.

FRANKEN, C. **Canário propõe medida de combate ao furto de animais na zona rural**. 2021. [Online; accessed 10-september-2021]. Disponível em: <<http://www.camarasap.rs.gov.br/sap/noticias/2695-canario-propoe-medida-de-combate-ao-furto-de-animais-na-zona-rural>>. Citado na página 17.

GAO, L. et al. Automatic learning of semantic region models for event recognition. In: **Proceedings - 8th International Conference on Intelligent Systems Design and Applications, ISDA 2008**. Kaohsiung, Taiwan: IEEE, 2008. v. 2, p. 40–44. Citado na página 26.

GHRAB, N. B.; FENDRI, E.; HAMMAMI, M. Abnormal events detection based on trajectory clustering. In: **Proceedings - Computer Graphics, Imaging and Visualization: New Techniques and Trends, CGiV 2016**. Beni Mellal, Morocco: IEEE, 2016. p. 301–306. Citado 5 vezes nas páginas 15, 26, 27, 29 e 30.

GONZALEZ, R. C.; WOODS, R. E. **Processamento de imagens digitais**. [S.l.]: Editora Blucher, 2000. Citado na página 33.

JIANG, F.; WU, Y.; KATSAGGELOS, A. K. Abnormal event detection based on trajectory clustering by 2-depth greedy search. In: **ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings**. Las Vegas, NV, USA: IEEE, 2008. p. 2129–2132. Citado 2 vezes nas páginas 24 e 26.

JURIĆ, D. **Rastreamento de objetos: Filtro Kalman com facilidade**. 2015. [Online; accessed 20-maio-2021]. Citado na página 35.

KIRAN, B. R.; THOMAS, D. M.; PARAKKAL, R. An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos. **Journal of Imaging**, Multidisciplinary Digital Publishing Institute, v. 4, n. 2, p. 36, 2018. Citado na página 15.

MAIYA, S. R. **DeepSORT: Deep Learning to Track Custom Objects in a Video**. 2019. [Online; accessed 20-outubro-2020]. Citado 2 vezes nas páginas 35 e 42.

MOHAMED, M. G.; SAUNIER, N. **Behavior analysis using a multilevel motion pattern learning framework**. [S.l.: s.n.], 2015. v. 2528. 116-127 p. (Transportation Research Record, v. 2528). Citado na página 26.

MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre aprendizado de máquina. **Sistemas inteligentes-Fundamentos e aplicações**, Manole, v. 1, n. 1, p. 32, 2003. Citado na página 33.

MORRIS, B. T.; TRIVEDI, M. M. Trajectory learning for activity understanding: Unsupervised, multilevel, and long-term adaptive approach. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, IEEE, v. 33, n. 11, p. 2287–2301, 2011. Citado 4 vezes nas páginas 26, 27, 29 e 30.

- MURUGAVEL, M. **Object Tracking — Referenced with the previous frame using Euclidean distance**. 2019. [Online; accessed 28-outubro-2020]. Citado na página 42.
- NAFTEL, A.; KHALID, S. Motion clustering using spatiotemporal approximations. In: **Proceedings of the IASTED International Conference on Internet and Multimedia Systems and Applications, IMSA**. [S.l.: s.n.], 2005. p. 207–212. Citado 4 vezes nas páginas 26, 30, 31 e 37.
- NAFTEL, A.; KHALID, S. Classifying spatiotemporal object trajectories using unsupervised learning in the coefficient feature space. **Multimedia Systems**, v. 12, n. 3, p. 227–238, 2006. Citado 2 vezes nas páginas 26 e 31.
- NAFTEL, A.; KHALID, S. Motion trajectory learning in the dft-coefficient feature space. In: **Proceedings of the Fourth IEEE International Conference on Computer Vision Systems, ICVS'06**. New York, NY, USA: IEEE, 2006. v. 2006, p. 47. Citado 2 vezes nas páginas 26 e 31.
- PACHECO, A. **O problema de otimização**. 2015. [Online; accessed 20-abril-2021]. Citado 2 vezes nas páginas 37 e 38.
- PALANISAMY, G.; MANIKANDAN, T. Group behaviour profiling for detection of anomaly in crowd. In: **2017 International Conference on Technical Advancements in Computers and Communications (ICTACC)**. Melmaurvathur, India: IEEE, 2017. p. 11–15. Citado na página 15.
- PALAZZO, S.; SPAMPINATO, C.; BEYAN, C. Event detection in underwater domain by exploiting fish trajectory clustering. In: **MAED 2012 - Proceedings of the 2012 ACM Workshop on Multimedia Analysis for Ecological Data, Co-located with ACM Multimedia 2012**. New York, NY, USA: ACM, 2012. p. 31–36. Citado 2 vezes nas páginas 26 e 27.
- PICIARELLI, C.; FORESTI, G. L. On-line trajectory clustering for anomalous events detection. **Pattern Recognition Letters**, v. 27, n. 15, p. 1835–1842, 2006. Citado na página 26.
- PUDASAINI, D.; ABHARI, A. Scalable pattern recognition and real time tracking of moving objects. In: **Simulation Series**. Tucson, AZ, USA: [s.n.], 2019. v. 51. Citado 3 vezes nas páginas 26, 27 e 30.
- PéTRIN, A. **Crimes no campo contam com a participação de funcionários e parentes**. 2021. [Online; accessed 10-july-2021]. Disponível em: <<https://www.canalrural.com.br/programas/informacao/rural-noticias/crimes-no-campo-contam-com-a-participacao-de-funcionarios-e-parentes/>>. Citado 2 vezes nas páginas 16 e 17.
- QAROUT, Y. K.; LOWE, D. 'surprise' detection in human pattern-of-life behaviour. In: **2018 5th International Conference on Information and Communication Technologies for Disaster Management, ICT-DM 2018**. [S.l.: s.n.], 2019. Citado na página 26.

QI, Y.; LIU, T.; FU, Y. Anomalous action recognition research for few-shot learning. In: **IEEE. 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)**. Chongqing, China: IEEE, 2020. v. 1, p. 1306–1310. Citado na página 15.

RABELO, E. **Flexibilidade: Um princípio de Arquitetura de Software**. 2019. [Online; accessed 25-july-2021]. Disponível em: <<https://oieduardorabelo.medium.com/flexibilidade-um-princC3ADpio-de-arquitetura-de-software-de1d9e2314bb>>. Citado na página 38.

RANJITH, R.; ATHANESIOUS, J. J.; VAIDEHI, V. Anomaly detection using dbscan clustering technique for traffic video surveillance. In: **ICoAC 2015 - 7th International Conference on Advanced Computing**. Chennai, India: IEEE, 2016. Citado 3 vezes nas páginas 26, 27 e 30.

REDMON, J.; FARHADI, A. Yolov3: An incremental improvement. **arXiv**, 2018. Citado na página 34.

RIBEIRO, C. **Segurança Pública Rural. Ribeiro, CNA SENAR**. 2019. Disponível em: <<https://www2.camara.leg.br/atividade-legislativa/comissoes/comissoes-permanentes/capadr/audiencias-publicas/audiencias-publicas-2019/audiencia-publica-12-de-setembro-de-2019-cna>>. Citado na página 16.

ROSEBROCK, A. **Simple object tracking with OpenCV**. 2018. [Online; accessed 30-outubro-2020]. Citado na página 42.

SAINI, R. et al. **Classification of object trajectories represented by high-level features using unsupervised learning**. Singapore: Springer, 2017. v. 459 AISC. 273-284 p. (Advances in Intelligent Systems and Computing, v. 459 AISC). Citado na página 26.

SANTOS, R. D. dos. **Pecuaristas têm prejuízos exorbitantes com furto de gado**. 2021. [Online; accessed 10-september-2021]. Disponível em: <<https://www.renatodiasdossantos.adv.br/pecuaristas-tem-prejuizos-exorbitantes-com-furto-de-gado/>>. Citado na página 17.

SILVA, C. D. M. da. **Lei 13.330/2016 criou nova forma do crime de furto, o de animais de produção**. 2016. [Online; accessed 10-july-2021]. Disponível em: <<http://www.sedep.com.br/artigos/lei-13-3302016-criou-nova-forma-do-crime-de-furto-o-de-animais-de-producao/>>. Citado 2 vezes nas páginas 16 e 17.

SINWAR, D.; KAUSHIK, R. Study of euclidean and manhattan distance metrics using simple k-means clustering. **International Journal for Research in Applied Science and Engineering Technology**, v. 2, n. 5, p. 270–274, 2014. Citado na página 45.

SOARES, T. O.; THIELO, M. R. Rastreamento de pessoas e animais em ambientes rurais. **Anais do Salão Internacional de Ensino, Pesquisa e Extensão**, v. 12, n. 2, 2020. Citado 2 vezes nas páginas 35 e 43.

Sodemann, A. A.; Ross, M. P.; Borghetti, B. J. A review of anomaly detection in automated surveillance. **IEEE Transactions on Systems, Man, and Cybernetics**,

**Part C (Applications and Reviews)**, v. 42, n. 6, p. 1257–1272, 2012. Citado 2 vezes nas páginas 36 e 37.

SWEARS, E.; HOOGS, A.; PERERA, A. G. A. Learning motion patterns in surveillance video using hmm clustering. In: **2008 IEEE Workshop on Motion and Video Computing, WMVC**. Copper Mountain, CO, USA: IEEE, 2008. Citado na página 26.

SZELISKI, R. **Computer vision: algorithms and applications**. Verlag London: Springer, 2010. Citado na página 33.

TAVARES, I. **Modelagem computacional em Física utilizando Python e VPython**. 2015. [Online; accessed 20-august-2021]. Disponível em: <<http://cta.if.ufrgs.br/attachments/1828/MUV.py>>. Citado na página 46.

THIELO, M. R. Análise e classificação de séries temporais não estacionárias utilizando métodos não-lineares. 2000. Citado na página 45.

TIMAN, A. F. **Theory of approximation of functions of a real variable**. [S.l.]: Elsevier, 2014. Citado na página 31.

WINCKLER, S. C.; THIELO, M. R. Detecção de eventos suspeitos no ambiente rural. **Anais do Salão Internacional de Ensino, Pesquisa e Extensão**, v. 12, n. 2, 2020. Citado 2 vezes nas páginas 16 e 41.

XIANG, T.; GONG, S. Video behaviour profiling and abnormality detection without manual labelling. In: . Beijing, China: IEEE, 2005. v. 2, p. 1238– 1245 Vol. 2. ISBN 0-7695-2334-X. Citado na página 36.

XU, M. et al. An efficient method of crowd aggregation computation in public areas. **IEEE Transactions on Circuits and Systems for Video Technology**, IEEE, v. 28, n. 10, p. 2814–2825, 2018. Citado na página 26.

XU, S. et al. Unsupervised abnormal behaviour detection with overhead crowd video. In: **International Conference on Software, Knowledge Information, Industrial Management and Applications, SKIMA**. Malabe, Sri Lanka: IEEE, 2018. v. 2017-December. Citado 4 vezes nas páginas 26, 27, 29 e 31.

YANG, M. et al. **Anomalous behavior detection in crowded scenes using clustering and spatio-temporal features**. Cham: Springer, 2016. v. 486. 132-141 p. (IFIP Advances in Information and Communication Technology, v. 486). Citado 2 vezes nas páginas 26 e 27.

ZANG, X. et al. An object-aware anomaly detection and localization in surveillance videos. In: **Proceedings - 2016 IEEE 2nd International Conference on Multimedia Big Data, BigMM 2016**. Taipei, Taiwan: IEEE, 2016. p. 113–116. Citado na página 26.

ZHANG, D. et al. Dynamic model behavior analysis of small groups based on particle video. In: **2013 International Conference on Wireless Communications and Signal Processing, WCSP 2013**. Hangzhou, China: IEEE, 2013. Citado 3 vezes nas páginas 26, 27 e 29.