

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
CURSO SUPERIOR DE ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

RODRIGO LUCIO DE LIMA

**AVALIAÇÃO DO ALGORÍTIMO SVM NA DETECÇÃO DE
COMPORTAMENTOS SUSPEITOS EM CENAS DE VÍDEO**

TRABALHO DE CONCLUSÃO DE CURSO

PONTA GROSSA

2014

RODRIGO LUCIO DE LIMA

AVALIAÇÃO DO ALGORÍTIMO SVM NA DETECÇÃO DE COMPORTAMENTOS SUSPEITOS EM CENAS DE VÍDEO

Trabalho De Conclusão de Curso apresentado ao DEPARTAMENTO ACADÊMICO DE INFORMÁTICA , para o Curso Superior De Análise e Desenvolvimento de Sistemas da Universidade Tecnológica Federal do Paraná como resultado de atividade de trabalho e pesquisa.

Orientador: Profº Geraldo Ranthum

PONTA GROSSA

2014



Ministério da Educação
**Universidade Tecnológica Federal do
Paraná**
Câmpus Ponta Grossa
Diretoria de Graduação e Educação
Profissional



TERMO DE APROVAÇÃO

Avaliação Do Algoritmo SVM Na Detecção De Comportamentos Suspeitos Em
Cenas De Vídeo.

por

RODRIGO LUCIO DE LIMA

Este Trabalho de Conclusão de Curso (TCC) foi apresentado em 26 de novembro de 2014 como requisito parcial para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Prof. Me. Geraldo Ranthum
Orientador

Prof. Me. Rogério Ranthum
Membro titular

Profª. Drª. Tânia Lúcia Monteiro
Responsável pelos Trabalhos
de Conclusão de Curso

Prof. Marcos Vinicius Fidelis
Membro titular

Profª. Drª Simone de Almeida
Coordenadora do curso

- O Termo de Aprovação assinado encontra-se na Coordenação do Curso -

AGRADECIMENTOS

Agradeço a Deus, soberano, que me criou dotado com inteligência para que pudesse concluir este trabalho. A meu pai que foi o responsável pela minha criação, pelo sustento, a quem devo grande respeito e admiração, por tudo que é, e pelo que fez.

A família que sempre com ternura me apoiou para que nunca desistisse, para que prosseguisse. Aos funcionários da instituição, que com notável empenho e dedicação nos proporcionam o ambiente nas condições necessárias para realizarmos nossas atividades. Ao professor Geraldo pela dedicação como professor orientador, sua disposição em ajudar e transmitir seus conhecimentos.

Agradeço também a minha namorada Eliane, por ser motivo de alegrias ímpares em minha vida.

RESUMO

LIMA, Rodrigo Lucio. **Avaliação Do Algoritmo SVM Na Detecção De Comportamentos Suspeitos Em Cenas De Vídeo**. 58 f. Trabalho De Conclusão de Curso – Curso Superior De Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2014.

A otimização de parâmetros no algoritmo *SVM* (*Support Vector Machines*), permite extrair os melhores resultados no contexto de *ML* (*machine Learning*). As Máquinas de Vetores de Suporte *SVM* estão entre os algoritmos de Aprendizado de Máquina que tem despertado o maior interesse desde sua introdução na área. Entretanto, para que tenha seu desempenho alcançado ao máximo deve ser estudado juntamente com a otimização de seus parâmetros. Este trabalho realizou experimentos, utilizando técnicas atuais com o objetivo de alcançar as melhores taxas de acertos no reconhecimento de movimentos suspeitos em cenas de vídeo. Através destes experimentos identificou-se que o método de atribuição de valores para seleção de parâmetros pode obter resultados similares aos obtidos em testes automatizados. Os métodos da literatura foram empregados nos experimentos e também métodos empíricos, isto sobre uma base de dados contendo informações sobre movimentos de pessoas em cenas de vídeo. Discute-se as particularidades das variações do algoritmo *SVM*, os resultados obtidos a cada etapa dos experimentos e os trabalhos sendo realizados com o algoritmo *SVM* em diversas áreas do *ML*. O resultado então pode ser comparado com as taxas obtidas em trabalhos anteriores.

Palavras-chave: *SVM*. Algoritmo. Comportamento. Suspeito.

ABSTRACT

LIMA, Rodrigo Lucio. ***SVM Algorithm Avaliation of Abnormal Behavior On Video Scenes*** . 58 f. Trabalho De Conclusão de Curso – Curso Superior De Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2014.

The parameters optimization in the SVM (Support Vector Machines) algorithm, allows to extract optimal results in the ML (Machine Learning) context. The Support Vector Machines take place between the ML algorithms that have received great interest since their introduction in the ML community. However, to achieve an optimal result it has to be studied along with it's parameters optimization. This work performed experiments , using atual tecnicas with the purpose of achieving optimal rates of suspect moviments recognition on vídeo scenes. Through this experiments have been proved that, value attribution parameter selection method can provide similar results as the automatized method can provide. The studied methods was applied and also the empirical methods, each one over an people moviments on video scenes data base. The variations of the SVM are discussed, the each step results achieved and the works being done with the SVM algorithm recently in ML. The result was able to be compared with those achieved in previous works

Keywords: SVM. Algorithm. Behavior. Abnormal.

LISTA DE FIGURAS

Figura 1 – Estrutura dos Sistemas de Vigilância	14
Figura 2 – Exemplo de Comportamentos	15
Figura 3 – Exemplo de Comportamento em Grupo	16
Figura 4 – Princípio SVM	18
Figura 5 – Exemplo Arquitetura SVM	19
Figura 6 – Exemplo de <i>Support Vectors</i>	19
Figura 7 – Exemplo de Hiperplano	21
Figura 8 – Exemplo <i>RBF</i>	23
Figura 9 – Estrutura de <i>Grid-search</i>	25
Figura 10- Exemplo de Faixa de Parâmetros	27
Figura 11- Exemplo de Parâmetro C	28
Figura 12- Exemplo de Parâmetro γ	29
Figura 13- Exemplos de Amostras Positivas e Negativas	31
Figura 14- Divisão da Região em Três Partes	33
Figura 15- Média de Erros	33
Figura 16- Fluxograma Algoritmo Comportamento Humano Agressivo	35
Figura 17- Exemplo de teste	41
Figura 18- Ações base de vídeos ICPR	43
Figura 19- Gráfico de Testes Não Automatizados	45
Figura 20- Exemplo de CVParameterselection	47
Figura 21- Gráfico de testes Sequencia única	49

LISTA DE TABELAS

Tabela 1 – Resultado do estudo :An Empirical Evaluation of Linear and Nonlinear Kernels for Text Classification Using Support Vector Machines	26
Tabela 2 – Comparação Final Método <i>Scene Specific</i> em relação á outros métodos	32
Tabela 3 – Acurácia(base de dados própria, boxe, com <i>BOF</i>).	36
Tabela 4 – Acurácia(base de dados <i>KTH</i> , boxe, com <i>BOF</i>).	36
Tabela 5 – Acurácia(base de dados própria, boxe, com Abordagem Geométrica).	36
Tabela 6 – Acurácia(base de dados <i>KTH</i> , boxe, com Abordagem Geométrica	37
Tabela 7 – Comparação de taxas de reconhecimento entre diferentes métodos base de dados <i>KTH</i>	37
Tabela 8 – Média Taxa de Acurácia	38
Tabela 9 – Média Taxa de Acurácia comparando com Estado da arte	39
Tabela 10– Porcentagens do melhor <i>kernel—kernel RBF</i>	44
Tabela 11– Comparação Final Testes Não Automatizados	45
Tabela 12– Comparação Final Bases de Dados Individuais	48
Tabela 13– Comparação Final Com Valores Exponenciais	48
Tabela 14– Testes Com Base Única	49
Tabela 15– Resultado Inicial x Resultado Final	50
Tabela 16– Resultados em Sequências Individuais	50

LISTA DE SIGLAS

<i>SVM</i>	<i>Support Vector Machines</i>
<i>ML</i>	<i>Machine Learning</i>
<i>C-SVC</i>	<i>C-Support Vector Classification</i>
<i>RBF</i>	<i>Radial Basis Function</i>
<i>LIBSVM</i>	<i>Library for Support Vector Machines</i>
<i>INRIA</i>	<i>Institut National de Recherche en Informatique et en Automatique</i>
<i>HOG-SVM</i>	<i>Histogram of Oriented Gradient -SVM</i>
<i>BOF</i>	<i>Bag of Features</i>
<i>MLP</i>	<i>Multy Layer Perception</i>
<i>SURF</i>	<i>Speeded Up Robust Features</i>
<i>PCA</i>	<i>Principal Component Analysis Algorithm</i>

SUMÁRIO

1	INTRODUÇÃO	10
1.1	OBJETIVOS	12
1.1.1	Objetivo Geral	12
1.1.2	Objetivos Específicos	12
1.1.3	Estrutura do Trabalho	12
2	REVISÃO BIBLIOGRÁFICA	14
2.1	ESTADO DA ARTE	14
2.2	DETECÇÃO DO MOVIMENTO	14
2.3	EXTRAÇÃO DE CARACTERÍSTICAS	15
2.4	CLASSIFICAÇÃO DO COMPORTAMENTO	16
2.4.1	Compreensão do Comportamento Humano	17
2.4.2	Entendimento do comportamento	17
2.5	<i>SVM SUPPORT VECTOR MACHINES</i>	17
2.5.1	C-SVC	20
2.5.2	Hiperplano	20
2.6	FUNÇÕES DE <i>KERNEL</i>	21
2.6.1	<i>Kernel RBF</i>	22
2.6.2	<i>Kernel Linear</i>	23
2.6.3	Função de <i>Kernel Polynomial</i>	23
2.6.4	<i>LIBSVM</i>	24
2.7	OTIMIZAÇÃO DE PARÂMETROS	24
2.7.1	<i>Grid-Search</i>	25
2.7.2	Faixa de valores de parâmetros	26
2.8	HIPERPARAMETROS	27
2.8.1	Parâmetro <i>C</i> (Cost)	27
2.9	PARÂMETROS DE <i>KERNEL</i>	28
2.9.1	Parâmetro γ (<i>Gamma</i>)	28
3	TRABALHOS CORRELATOS	30
4	MÉTODO PROPOSTO	40
4.1	METODOLOGIA DOS TESTES	40
4.1.1	Ambiente <i>Weka</i>	41
4.1.2	Bases de Vídeos Utilizadas	42
4.1.3	Testes Não Automatizados	43
4.1.4	Resultados de Testes Não Automatizados	44
4.1.5	Análise dos resultados dos Testes Não Automatizados	45
4.2	TESTES AUTOMATIZADOS	46
4.2.1	Alteração nos atributos usados	46
4.2.2	<i>Grid-Search</i> Bases de Dados Individuais com valores lineares	46
4.2.3	Análise dos Resultados <i>Grid-Search</i> Bases de Dados Individuais, com valores lineares	47
4.3	<i>GRID-SEARCH</i> COM VALORES EXPONENCIAIS	48

4.4 TESTES COM BASE ÚNICA.....	49
4.5 RESULTADO FINAL	49
4.6 DISCUSSÕES FINAIS	50
5 CONCLUSÃO E TRABALHOS FUTUROS	52
5.1 CONCLUSÃO.....	52
5.2 TRABALHOS FUTUROS.....	53
REFERÊNCIAS	55

1 INTRODUÇÃO

Com a crescente aplicação de câmeras de vídeo nos sistemas de vigilância, as novas tecnologias de análises de vídeos tem atraído a atenção na área de Visão Computacional, e faz da compreensão do comportamento humano um importante tópico para o desenvolvimento de pesquisas.

Tal compreensão em análise do comportamento humano, em especial os comportamentos anormais, como por exemplo agressão física, proporcionaria a prevenção de crimes e permitiria menor tempo de resposta para as autoridades legais (XU et al., 2010).

A compreensão do comportamento humano visa estabelecer uma descrição semântica para ações e interações humanas através da análise e reconhecimento de seus padrões de movimentos. Esta semântica é ilustrada por (KAUTZ, 1987) da seguinte forma: para fazer uma caminhada uma pessoa pode ir para um bosque, para caçar outra pessoa pode também ir a um bosque.

Para caçar uma pessoa pode pegar uma arma, e para roubar um banco uma pessoa também pode pegar um arma. Ou seja, para um mesmo padrão de movimento, pode existir mais de um significado. O que torna o desafio da compreensão ainda maior.

O processo de compreensão do comportamento humano inicia-se quando as informações visuais são extraídas de cenas de vídeos, e representadas em forma de características, e estas características precisam ser relevantes ao contexto. estas após serem extraídas precisam então ser comparadas com características já rotuladas e que representam típicos comportamentos de seres humanos.

A pesquisa nesta área pode se dividir em duas perspectivas, a do comportamento humano normal, e a do comportamento humano anormal. Os dados sobre comportamento humano anormal são bastante escassos em relação ao comportamento humano normal. Desta forma, faz-se necessário que mesmo os padrões de movimentos

normais estejam ajustados de acordo com sua aplicação específica.

Por exemplo no contexto de um sistema de vigilância por câmeras, deve haver uma lista de movimentos considerados normais que são usados para treinar o sistema. Enquanto os movimentos anormais, não podem ser encontrados nesta lista, por serem difíceis de prever.

Em uma situação específica, uma pessoa sendo monitorada pode mudar seu comportamento anormal, depois de ter seu perfil mapeado. Isto pode levar a um falso alarme. Tendo então o sistema que ser novamente treinado.

Diversos algoritmos são utilizados visando avaliar dados referentes a diversas situações, dentre as quais o movimento humano. Neste contexto, o algoritmo *SVM* proporciona uma forma de encontrar o melhor resultado no reconhecimento de movimentos suspeitos em cenas de vídeos. *SVM* são um grupo de métodos de aprendizado de máquina, amplamente utilizado devido a suas atrativas características e desempenho de generalização promissora. Algumas destas características são a habilidade em modelar problemas não lineares, e que, sua capacidade de generalização não depende da dimensão do espaço.

No entanto um fator limita a aplicação do *SVM*, que é a otimização de seus parâmetros, ou dos parâmetros da função de *kernel*. Sendo assim, um dos objetivos deste trabalho é a otimização de parâmetros, visto que, desta maneira é possível maximizar os resultados do *SVM*, mas esta é uma tarefa complexa, pois exige um conhecimento prévio dos tipos de *kernel*, técnicas aplicáveis e não havendo uma técnica razoável que auxilie no processo, esta tarefa pode demandar um alto custo em termos de tempo. Durante o processo também é preciso levar em consideração que, a otimização, e a correta parametrização tem influência direta no desempenho de generalização do algoritmo.

Os experimentos realizados neste trabalho visam também a minimização do tempo despendido na seleção dos parâmetros. Tendo-se em vista que, são mais de um parâmetro por *kernel*, e estes podendo variar em valores baixos a cada iteração, o custo em termos de tempo pode ser alto, tornando inviável tal otimização.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

- O objetivo do trabalho constitui melhorar taxas de acertos no reconhecimento de comportamento suspeitos nas imagens de vídeo.

1.1.2 Objetivos Específicos

- Avaliação do comportamento de diferentes tipos de *kernel* no reconhecimento de comportamentos suspeitos em imagens de vídeo.
- Realização de *Grid-Search* não automatizado, para comparação com *Grid-Search* automatizado.
- Avaliação de desempenho do *Grid-Search* no ambiente *Weka*.
- Avaliação do impacto da otimização dos parâmetros do *SVM*.

1.1.3 Estrutura do Trabalho

Este trabalho estruturado como segue:

No capítulo um são dadas informações a respeito do aspecto geral do trabalho e suas possíveis aplicações. Aspectos como a crescente utilização de Sistemas de Monitoramento e suas deficiências, possíveis soluções, como as tratadas neste trabalho. É enfatizada a complexidade da compreensão do comportamento humano em especial quando se trata da detecção de comportamentos anormais. Também é abordado a respeito das variáveis envolvidas na otimização de parâmetros e as dificuldades encontradas.

No capítulo dois é abordado o estado da arte como a estrutura dos sistemas de vigilância, detecção de movimentos, compreensão do comportamento humano, e sobre o algoritmo *SVM*, seus tipos de *kernel* e sobre a otimização de parâmetros.

No capítulo três os trabalhos correlatos onde serão apresentados trabalhos na área de aprendizagem de máquina utilizando o *SVM*. Foram tratados trabalhos recentes, visando situar o leitor nas técnicas e pesquisas atuais, em contexto com as referências clássicas a respeito do *SVM*.

No capítulo quatro é abordado o método proposto onde será apresentado o método utilizado na realização deste trabalho. É descrito o método de atribuição de valores, o método utilizando *grid-search* e também as recomendações de vários autores para otimização de parâmetros. O ambiente Weka é também alvo de análises quanto a seu desempenho em relação às recomendações de autores e também quanto às expectativas durante a realização do trabalho, as bases de dados que originaram as bases utilizadas neste trabalho são contextualizadas .

Finalizando, o Capítulo cinco, apresentará as conclusões obtidas á partir da realização dos experimentos e as contribuições esperadas a partir do término deste trabalho. Apresentará ainda quais os problemas não solucionados neste trabalho. Também fará uma sugestão para futuras pesquisas que poderão ser realizadas sobre o tema.

2 REVISÃO BIBLIOGRÁFICA

2.1 ESTADO DA ARTE

A vigilância visual em cenas dinâmicas tenta detectar, reconhecer e rastrear objetos na sequência de imagens, além de compreender e descrever os seus comportamentos. No trabalho apresentado por (HU et al., 2004) foi proposta uma estrutura genérica para os sistemas de vigilância, apresentada na Figura 1.

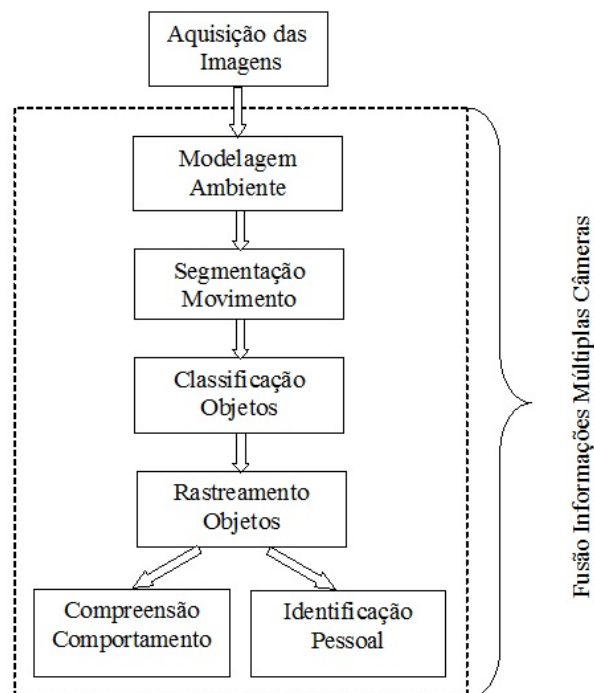


Figura 1: Estrutura dos Sistemas de Vigilância

Fonte: (HU et al., 2004)

2.2 DETECÇÃO DO MOVIMENTO

A maioria dos sistemas de vigilância visuais iniciam-se com a detecção do movimento. As fases seguintes, como rastreamento e reconhecimento de comportamento

dependem diretamente da detecção do movimento. Usualmente o processo de detecção de movimento envolve modelagem do ambiente e segmentação do movimento. Os modelos de ambiente podem ser classificados em 2D numa imagem plana, e 3D em coordenadas reais. A modelagem 2D devido a simplicidade tem muito mais aplicações. A segmentação do movimento tem como objetivo detectar regiões correspondentes a objetos em movimento, assim como veículos e seres humanos (HU et al., 2004).

2.3 EXTRAÇÃO DE CARACTERÍSTICAS

A extração de características (JAIN et al., 2000) visa reduzir a dimensão dos dados e deve identificar um adequado subconjunto de características para representar um padrão a partir de um conjunto maior, que pode apresentar características irrelevantes ou redundantes. Após selecionadas, apenas características relevantes serão analisadas para o domínio da aplicação, o classificador seria mais rápido e necessitaria de menos memória. Além disso, estas características mantêm a sua interpretação física original e as propriedades de quando foram criadas.

A extração de características é uma tarefa determinante, pois consegue-se inferir conhecimento sobre o comportamento do objeto, possibilitando detectar se este comportamento é anormal para aquele contexto. Segundo (WU et al., 2013, p. 214)“...além de se escolher as melhores características, é importante escolher as características que proporcionem resultado superior”. A figura 2 ilustra como características de espaço tempo da ação por exemplo, de correr, podem levar a diferentes comportamentos, dependendo do contexto.

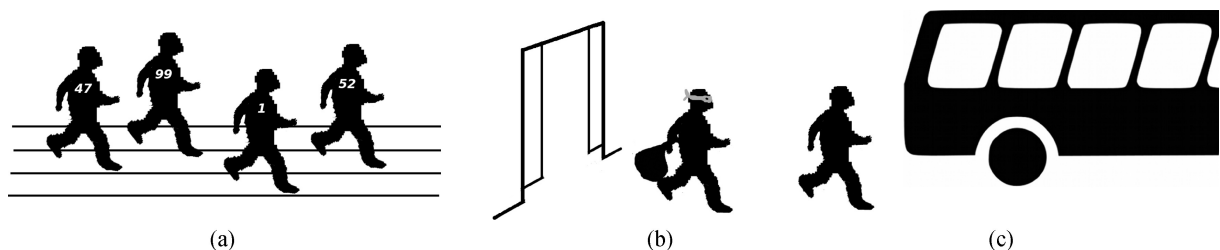


Figura 2: Exemplo de Comportamentos gerados por similares características de espaço tempo, (a) Esportes, (b) Assalto, (c) Embarque em Ônibus.

Fonte: (BORGES et al., 2013)

2.4 CLASSIFICAÇÃO DO COMPORTAMENTO

Bouzegza e Elarbi-Boudiher (2013, p. 189) aponta que “...um dos principais focos de pesquisadores desde setembro de 2001, é a vigilância automática por vídeos, que tem como alvo detectar automaticamente o comportamento humano anormal”. O comportamento de grupos de pessoas é desafiador, com intensas pesquisas na área. A classificação do comportamento humano sofre muitas interferências, quando se fala de vigilância, a dinâmica do ambiente, luminosidade, a limitação do ser humano em monitorar vários *streams* de vídeos.

Em muitos casos quando se fala em comportamento em grupo, estes grupos são compostos por um ou mais sub-grupos, e as atividades destes sub-grupos normalmente não serão as mesmas, como: aproximações, divisões, pessoas sendo seguidas etc. , para (ZHANG et al., 2012, p. 44), “...isto torna a detecção de comportamentos em grupo um desafio”. A figura 3 ilustra as atividades de grupos e sub-grupos.

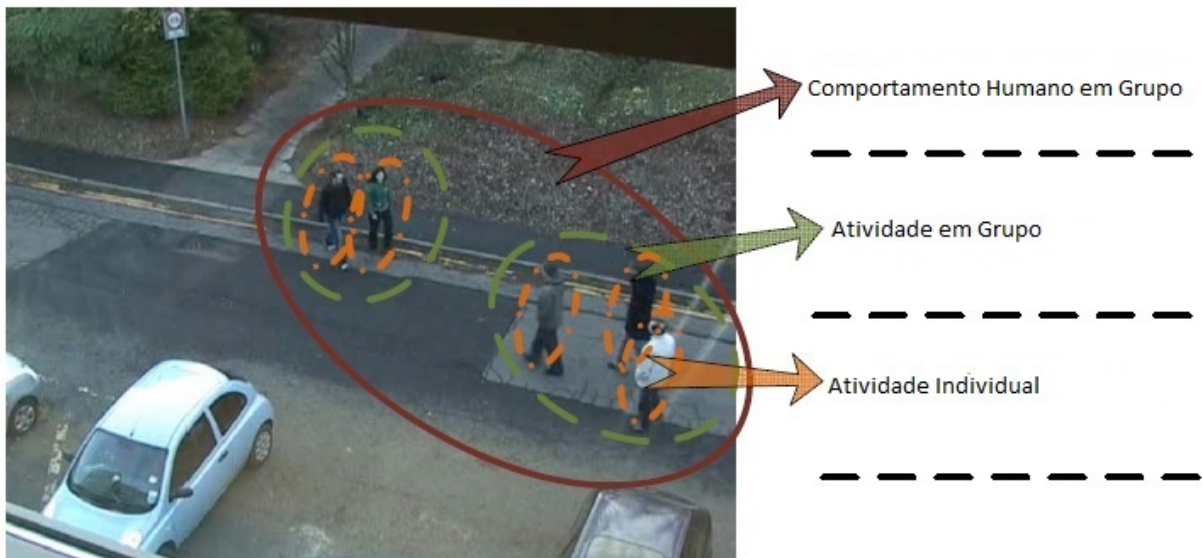


Figura 3: Exemplo de Comportamento em Grupo ilustrando as Atividades de Sub-Grupos

Fonte: (ZHANG et al., 2012)

O estabelecimento de atividades ou um modelo para representar atividades em grupo é a chave para o reconhecimento do comportamento. Entretanto estas dificuldades podem ser superadas através de técnicas de reconhecimento de comportamentos.

2.4.1 Compreensão do Comportamento Humano

Com a crescente aplicação de câmeras de vídeo nos sistemas de vigilância, as novas tecnologias de análises de vídeos tem atraído a atenção na área de Visão Computacional, e faz da compreensão do comportamento humano uma importante direção para pesquisas. Tal compreensão em análise do comportamento humano, em especial os comportamentos anormais, como por exemplo agressão física, proporcionaria a prevenção de crimes e permitiria menor tempo de resposta para as autoridades legais (XU et al., 2010).

2.4.2 Entendimento do comportamento

Após a segmentação e rastreamento de seres humanos em cenas de vídeo, o reconhecimento é o próximo passo para o entendimento do comportamento humano. Segundo (XU et al., 2010, p. 3869), "... o entendimento do comportamento consiste em fornecer uma descrição semântica para as ações e interações humanas através de seus padrões de movimento". Durante a classificação, duas das questões relevantes que precisam ser levadas em conta são: obter as sequências de referência de comportamentos dos conjuntos de treinamento e realizar o treinamento com métodos que proporcionem o menos desvio nas escalas de espaço e tempo para padrões similares de comportamento.

2.5 SVM SUPPORT VECTOR MACHINES

SVM's desenvolvido por (VAPNIK, 1995), são um grupo de métodos de aprendizado de máquina usados para classificação e regressão, baseia-se no conceito de planos de decisão que definem os limites de decisão, estes planos separam objetos de diferentes classes (VIDHYA, 2013).

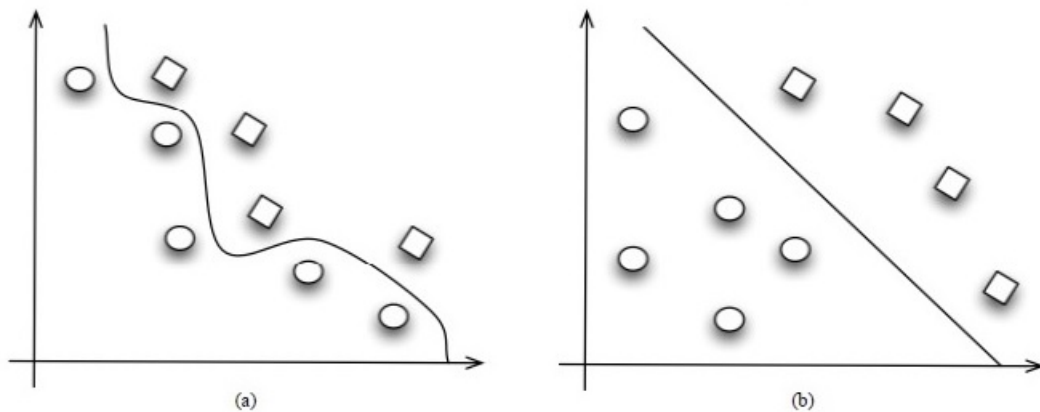
Estes métodos vem produzindo excelentes resultados em diversos problemas de classificação, como o reconhecimento de caracteres manuscritos, detecção de faces e imagens, categorização de texto, análises de bio-sequencias e segundo (PRIYANKA; GUPTA, 2013, p. 105) "... o *SVM* vem produzindo os melhores resultados na análise de sentimentos ". Para (NOVAKOVIC; VELJOVIC, 2011, p. 465) "...para diagnósticos médicos, o *SVM* é popular devido a sua excelente generalização. "

Uma das principais características do classificador *SVM* é a robustez em grandes

dimensões, possibilitando a sua aplicação em vetores de características de grandes dimensões, por exemplo, contendo imagens, como as analisadas neste trabalho. O SVM consiste em uma máquina de aprendizado estático, implementa o princípio de Minimização de Risco Estrutural (*SRM, structural risk minimization*) construindo um hiperplano ótimo que maximiza a margem de separação entre diferentes classes de dados, no caso de problemas lineares. “...Devido ao seu bom desempenho, SVM tornou-se um referencial na pesquisa em *ML*, ou aprendizagem de máquina ” (SOLIMAN; MAHMOUD, 2012, p. 182).

Para problemas não lineares o algoritmo usa funções de *kernel* para mapear o espaço de entrada, que é de alta dimensão, e o espaço chamado de espaço de características, onde as classes são linearmente separáveis e “... desta forma encontra uma solução única ” (LIU; JIANG, 2008).

No SVM, a ideia principal é mapear o espaço original (x), em um espaço de características (f), de alta dimensão através de uma função de mapeamento não linear, conforme mostrado na figura 4 . O SVM é uma técnica desenvolvida a cerca de 20 anos , (JIN et al., 2011, p.3667) afirma que “... o SVM como por ser uma técnica recém desenvolvida, existem ainda pontos a serem desenvolvidos”.



**Figura 4: a)Exemplo Espaço de Entrada onde as classes não são linearmente separáveis
b) Espaço de Características onde as classes são linearmente separáveis.**

Fonte: (SOTOMAYOR et al., 2013)

A figura 5 ilustra a arquitetura do SVM.

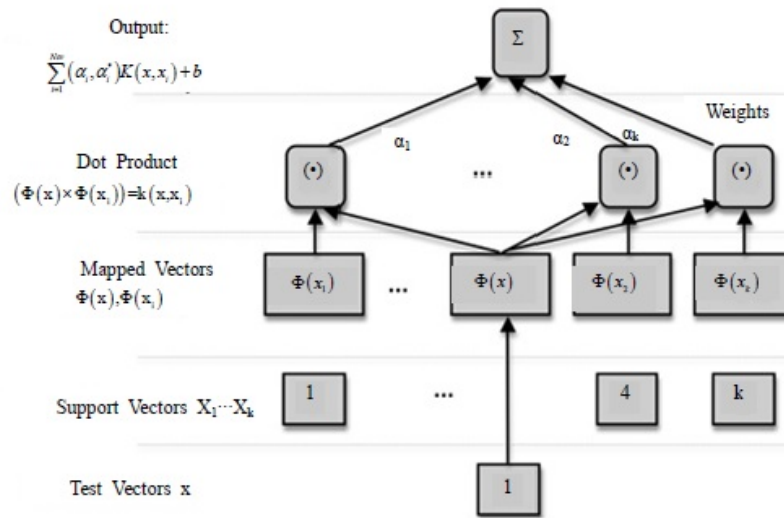


Figura 5: Exemplo Arquitetura SVM

Fonte: (SOTOMAYOR et al., 2013)

A figura 6 ilustra graficamente o que são os *Support Vectors*, que são os vetores mais próximos ao limite da dos hiperplanos de suporte, ou seja os vetores destacados na imagem.

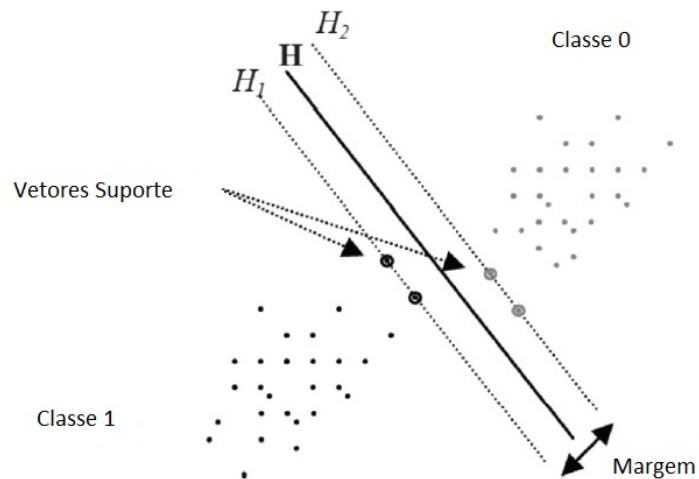


Figura 6: Os Vetores de Suporte são os pontos tocados pelas linhas tracejadas

Fonte: (TICAY et al., 2012)

2.5.1 C-SVC

Há diferentes tipos de *SVM* e para este estudo foi escolhido, com base em bons resultados na área de pesquisa o *C-SVC* (*C-Support Vector Classification*) (NOVAKOVIC; VELJOVIC, 2011). *C-SVC* pode ser usado com diferentes tipos de *kernel*, e ainda segundo (NOVAKOVIC; VELJOVIC, 2011, p. 466) "... a escolha de determinado *kernel* para um trabalho ou aplicação específica frequentemente é uma tarefa difícil, pois não há uma estrutura matemática que indique a melhor opção ". Espera-se que um *kernel* não linear baseado em *C-SVC* tenha melhor performance sobre um baseado em *kernel linear*.

2.5.2 Hiperplano

O *SVM* usa um hiperplano, ou um conjunto de hiperplanos para dividir o espaço em zonas, de forma que tais zonas contenham classes comuns. Rotulando estas zonas o sistema é capaz de identificar seus elementos através de uma amostra. O hiperplano é onde ocorre a separação das instâncias. O *SVM* seleciona todas as amostras críticas (*Support Vectors*) e constrói uma função linear que as distingue. Isto torna conforme já citado, a separação não linear possível.

"...O aspecto mais interessante do *SVM* é a capacidade de fazer a separação das instâncias quando as classes não são linearmente separáveis. Isto porque o algoritmo projeta o problema em um espaço de alta dimensão, tornando as classes linearmente separáveis " (RAJKUMAR; JAGANATHAN, 2013, p. 554).

No caso deste trabalho o hiperplano será responsável por fazer a separação dos comportamentos suspeitos dos comportamentos considerados comuns. A figura 7 ilustra um hiperplano de separação, que é um dos conceitos mais importantes dentro do *SVM*.

Conforme a figura 7, o hiperplano com a máxima margem foi construído tendo o suporte de dois outros hiperplanos, ou hiperplanos de suporte, estes por sua vez passam por alguns pontos das duas classes, que são os *Support Vectors* (Vetores Suporte).

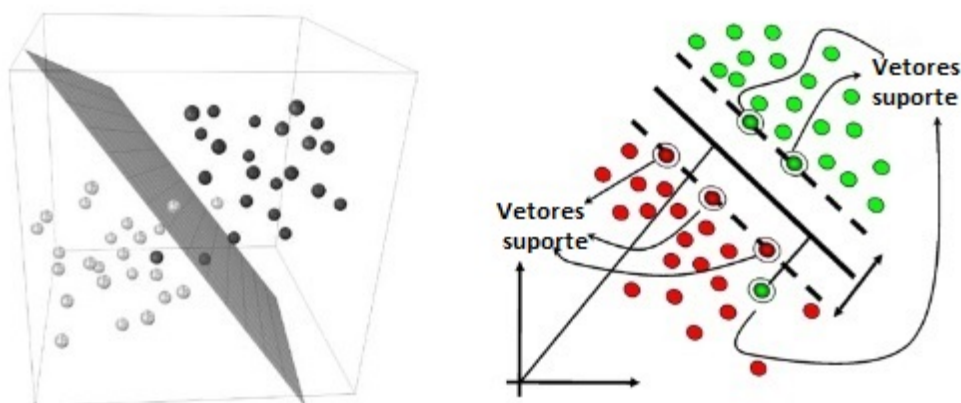


Figura 7: Princípios do SVM : Exemplo de Hiperplano(linha contínua no centro) e Hiperplanos de Suporte(linhas tracejadas nas laterais)

Fonte: (SCHOLTEN et al., 2012)

2.6 FUNÇÕES DE *KERNEL*

Funções de *Kernel* são recursos do SVM, muito úteis por proverem uma ponte simples entre algoritmos lineares e não lineares. A função de *kernel* é um artifício matemático que permite ao SVM executar uma classificação bi dimensional de uma base de dados que originalmente é de uma dimensão. SVM usa as funções de *kernel* para mapear as classes não separáveis linearmente, do espaço original, para o espaço do *kernel*.

“... Um *kernel* é utilizado pelo SVM para minimizar o risco empírico e a complexidade do modelo” (LEIGHTLEY et al., 2013, p. 262), e é no espaço do *kernel* que as classes não separáveis são separadas pelo hiperplano. A seleção dos parâmetros do *kernel* tem influência direta no desempenho do aprendizado de máquina, e “... o parâmetro de penalidade C e os parâmetros de *kernel* são difíceis de escolher e ainda não chegou se a um método que proporcione um resultado ótimo” (BAI et al., 2012, p. 113).

Segundo Liu e Jiang (2008), apesar de o SVM obter sucesso numa vasta área de aplicação, um problema notório aparece na sua aplicação prática. Esse problema inclui

principalmente o parâmetro de penalidade C (*Cost*), o parâmetro da função de perda (*Loss*), e os parâmetros na função de *kernel* (por exemplo, parâmetros de largura da função de *kernel RBF* (*Radial Basis Function*), e seu efeito para mais ou para menos na performance do *SVM*.

A função de perda (*Loss*) é definida na equação (1).

$$L(f(x), y) = \max\{|f(x) - y| - 3, 0\} \quad (1)$$

2.6.1 *Kernel RBF*

O *RBf* (*Radial Basis Function*) é bastante utilizado para resolução de problemas de aprendizado de máquina. O *RBf* se diferencia do *kernel linear*, pois nele é possível resolver problemas que originalmente não são linearmente separáveis, através do mapeamento para um espaço de alta dimensão. Existem dois parâmetros que podem ser explorados na busca de um melhor resultado para o aprendizado, que são: γ (*gamma*) e C (*Cost*) (VIDHYA, 2013).

Em grande parte dos problemas de classificação, o *kernel RBF* obtém um desempenho superior ao *kernel Linear* ou o *kernel Polynomial*, e por ter um reduzido número de parâmetros é tido como a primeira opção dentre os tipos de *kernel*. No entanto "...este *kernel* é difícil de modelar, no sentido de que é difícil de se chegar a um valor ótimo para γ (*gamma*), e escolher um valor correspondente para o parâmetro C (*Cost*) que funcione adequadamente para um determinado problema" (NOVAKOVIC; VELJOVIC, 2011, p. 466). A função que define o *kernel RBF* é mostrada na equação (2).

$$k(x, y) = \exp(-\gamma \langle x - y, x - y \rangle)^2 \quad (2)$$

A figura 8 ilustra a aplicação do *kernel RBF*.

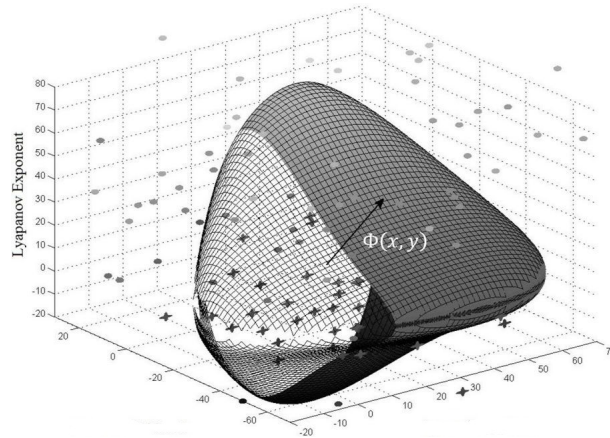


Figura 8: Exemplo de aplicação do *Kernel RBF*

Fonte: (ASADPOUR MOHAMMD REZA RAVANFAR, 2013)

2.6.2 *Kernel Linear*

É o *kernel* mais simples. Apresenta boa performance em problemas com dados linearmente separáveis, e surpreendentemente funciona bem com dados não separáveis linearmente. Esta informação será avaliada durante as avaliações realizadas neste trabalho. A equação que define o *kernel Linear* é mostrada na equação (3). O *SVM* linear usa uma função de *kernel dot product* e separa o espaço de características com um hiperplano (VETON, 2011).

$$K(x, y) = \langle x, y \rangle \quad (3)$$

2.6.3 Função de *Kernel Polynomial*

A função de *kernel Polynomial* é uma função de *kernel* global, com características globais, e cada um dos pontos da amostra tem impacto no valor da função de *kernel*. A função de *kernel polynomial* será linear quando a ordem for igual a 1. A função de *kernel Polynomial* é mostrada na equação (4)

$$K(x, y) = (\langle x, y \rangle + 1)^d \quad (4)$$

2.6.4 LIBSVM

O *LIBSVM*, (*Library for Support Vector Machines*) é uma biblioteca utilizada em problemas de classificação e regressão, desenvolvida por (CHANG; LIN, 2011). As principais características são: suporte a problemas multiclasse, treinamento com os tipos de *kernel* mais comuns, validação cruzada aplicada a escolha do melhor modelo e código fonte em C++ e Java. Por padrão, ela utiliza a função de *kernel RBF* e os parâmetros C (custo) e γ (gamma), são escolhidos segundo o problema a ser resolvido. Ao parâmetro C é atribuído o valor 1, e a γ atribuído um valor variável; esse valor depende do número de instâncias do domínio, sendo N esse número, γ seria igual a $1/N$.

A biblioteca *LIBSVM* (CHANG; LIN, 2011) disponibiliza todos os tipos de *kernel* mais comuns, incluindo o *kernel sigmóide* e o *RBF*; gera apenas um arquivo com resultados a cada treinamento e possibilita ao usuário gerenciar o espaço de memória requerido para a realização do treinamento. Todas essas características determinaram a escolha da biblioteca *LIBSVM* para a realização dos experimentos descritos neste trabalho.

O algoritmo *LIBSVM* foi utilizado neste trabalho, empregando-se as técnicas de *multifold-cross-validation* (onde o número de subconjuntos K foi igual a 10) e *percentage-split* (proporção 2/3 para treinamento e 1/3 para teste) base a base, e cada resultado, juntamente com seus parâmetros, foram armazenados em um arquivo, para posteriormente serem carregados em uma estrutura de dados para análise.

2.7 OTIMIZAÇÃO DE PARÂMETROS

A otimização de parâmetros é frequentemente uma tarefa com alto custo de tempo. E requer um conhecimento prévio sobre a base de dados, além disso, é preciso executar vários testes até que se encontrem os parâmetros ideais para dado algoritmo de classificação.

Neste contexto a meta que se espera alcançar com esta otimização é encontrar um grupo de parâmetros que maximize o resultado do algoritmo de classificação. Estes parâmetros tem influência direta no desempenho do classificador.

Segundo (REN; BAI, 2010, p. 1160) "...a aplicação prática do *SVM* tanto numa plataforma acadêmica quanto industrial é imitada porque a qualidade do modelo depende

profundamente da seleção adequada dos parâmetros do *SVM* e dos parâmetros da função de *kernel*".

2.7.1 Grid-Search

(HUANG; WANG, 2006, p. 236) define um algoritmo de *grid-search* como "... um método bastante utilizado, nele de C e γ são testados sucessivamente até que um par com o melhor resultado final seja encontrado. Então estes são escolhidos".

A figura 9 ilustra o comportamento de um *grid-search*, onde testes sucessivos na busca do par de parâmetros ótimos são realizados, a base de dados é dividida em um conjunto de testes e um conjunto de treinamento. O algoritmo é executado e os valores possíveis para os parâmetros são testados.

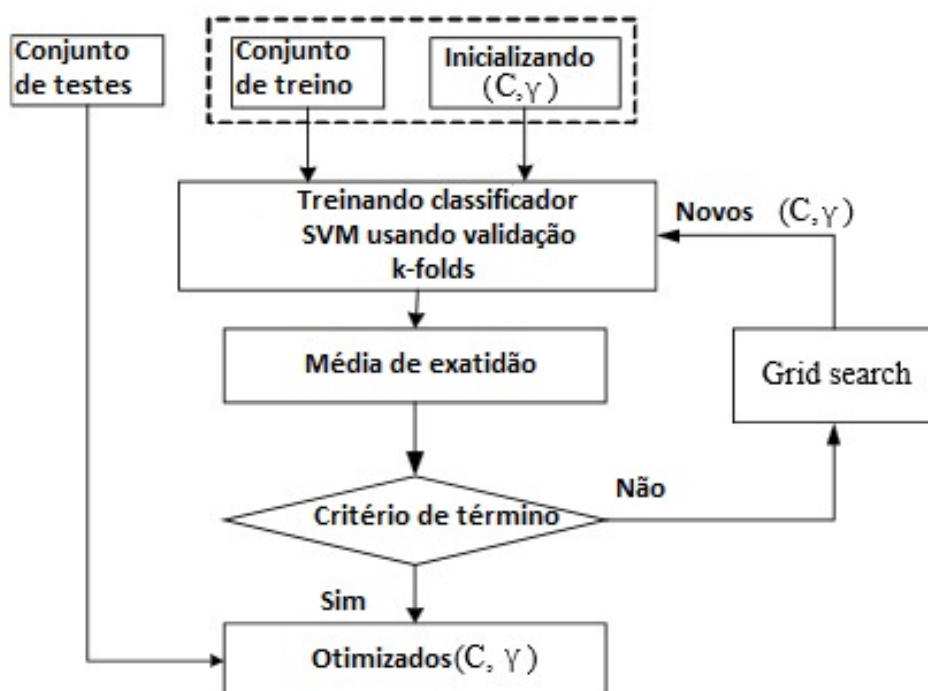


Figura 9: Ilustração do Ciclo de um Grid-Search na busca de Parâmetros Ótimos

Fonte: (HUANG; WANG, 2006)

Hsu et al. (2010) recomenda um *grid-search* (busca pelos melhores parâmetros de forma automatizada), utilizando sequências exponenciais. Isto baseado em resultados positivos obtidos em estudos anteriores, variando ($C = 2^{-5}, 2^{-3}, \dots, 2^{15}$) e ($\gamma = 2^{-15}, 2^{-13}, \dots, 2^3$). Segundo (JIN et al., 2011) alguns valores são recomendados para o *grid-search* usando *kernel RBF*, que seriam: ($C = 2^{25}, 2^{27}, \dots, 2^{35}$) e ($\gamma =$

$2^{-9}, 2^{-7}, 2^{-1}, 2^1, \dots, 2^7$).

Em um estudo realizado por (GAO; SUN, 2010, p. 1504) sobre classificação de páginas *web* usando *SVM* não linear, os valores utilizados para *grid-search* foram: 2^{-14} á 2^{14} .

Os resultados deste estudo aparecem na tabela 1 e nota-se que o *kernel* com melhor desempenho foi o *RBF*.

Tabela 1: Resultado do estudo :An Empirical Evaluation of Linear and Nonlinear Kernels for Text Classification Using Support Vector Machines

	Forma do Resultado da Matriz	% Precisão I	% Precisão II
Linear	(C, precisão)	95,9087	95,1457
Poly	(gamma, r, d,C, precisão)	96,0990	95,2426
Rbf	(gamma, C, precisão)	96,4795	95,3378
Sigmoid	(gamma, r,C, precisão)	96,0040	95,2426

As razões para o uso do *grid-search* são basicamente duas:

- Confiança de que as opções de valores foram testadas. A segurança nos resultados obtidos é maior, sabendo se que todos os valores foram testados e o processo foi automatizado.
- O Custo operacional de um *grid-search* não é mais alto do que usando métodos mais avançados, pois são dois parâmetros a serem testados.

2.7.2 Faixa de valores de parâmetros

Um fato interessante que se pode notar em um *grid-search*, é que existe uma área de valores de parâmetros que proporciona um desempenho ótimo para o classificador. Num estudo realizado por (GAO et al., 2013), para o parâmetro γ , sua faixa de valores foi dividida da seguinte forma: (0,1) em 20 partes iguais, com um espaçamento de 0.05 e notou se que quando $\gamma = 0.005$ e $\gamma = 0.1$ os resultados foram os mesmos e inevitavelmente quando $\gamma > 0.1$ o resultado caía.

“... Supondo-se que temos um valor para (γ, C). Se diminuirmos o valor de γ , isto diminui a curvatura do limite de decisão, e se aumentarmos o valor de C o limite de decisão é forçado a curvar-se para acomodar a alta penalidade de erros” (BEN-HUR; WESTON, 2008, p. 234).

Este comportamento pode ser visto na figura 10.

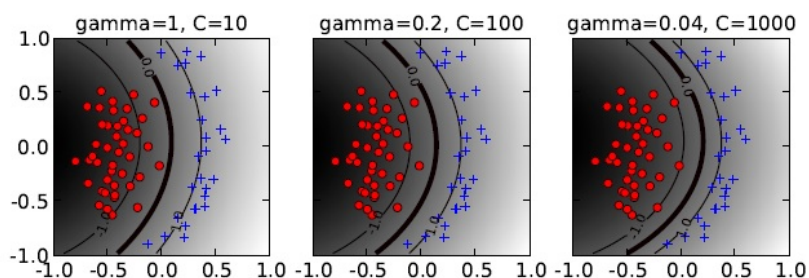


Figura 10: Semelhantes limites de decisões podem ser obtidos usando diferentes combinações de hiperparâmetros.

Fonte: (BEN-HUR; WESTON, 2008)

2.8 HIPERPARAMETROS

2.8.1 Parâmetro C (Cost)

C (*Cost*) determina o equilíbrio entre permitir erros de treinamento e forçar margens rígidas para permitir flexibilidade na separação das classes. Isto cria uma margem suave que permite alguns erros de classificação.

Aumentando-se o valor de C aumenta-se o custo do erro de classificação, então os dois pontos mais próximos ao hiperplano interferem em sua orientação e o hiperplano aproxima-se de vários outros pontos. Este comportamento é ilustrado na figura 11 (no painel da esquerda).

Se diminuirmos o valor de C (painel da direita) aqueles pontos tornam-se margens de erro; o hiperplano muda de orientação, e provê uma margem maior (BEN-HUR; WESTON, 2008).

Segundo (KOU et al., 2010, p. 1) o parâmetro de regularização C "... é usado para proporcionar equilíbrio entre erros no treinamento e complexidade do modelo, mas não tem um propósito específico, portanto é difícil definir seu valor".

“...O parâmetro C (*Cost*) realoca o ponto (instância de cada classe), mais próximo ao centro da sua classe”(SCHOLTEN et al., 2012, p. 4).

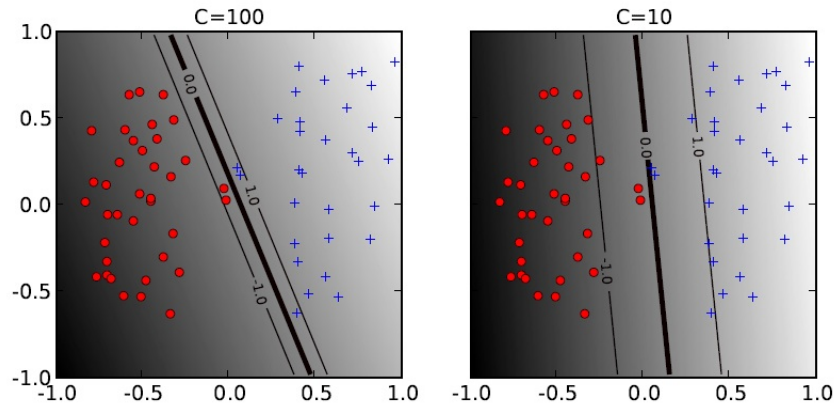


Figura 11: Exemplo de comportamento no parâmetro *Cost*. Penalidade do erro:distância do hiperplano, multiplicada pelo custo do erro C .

Fonte: (BEN-HUR; WESTON, 2008)

2.9 PARÂMETROS DE *KERNEL*

2.9.1 Parâmetro γ (*Gamma*)

Os parâmetros de *kernel* também tem significativo efeito nos limites da decisão. O parâmetro de largura do *kernel RBF* controla a flexibilidade do classificador.

Quando γ é baixo (no topo esquerdo do painel na figura 12) um dado ponto x tem um valor de *kernel* não zero para qualquer outro exemplo dos vetores de suporte. Portanto todo o conjunto de vetores de suporte afetam o valor da função em x , resultando em um limite de decisão mais macio (BEN-HUR; WESTON, 2008).

Na função $k(x,y) = \exp(-\gamma \langle x-y, x-y \rangle)^2$, o resultado é praticamente zero, se a distância entre x e x' é maior que $\frac{1}{\sqrt{\gamma}}$.

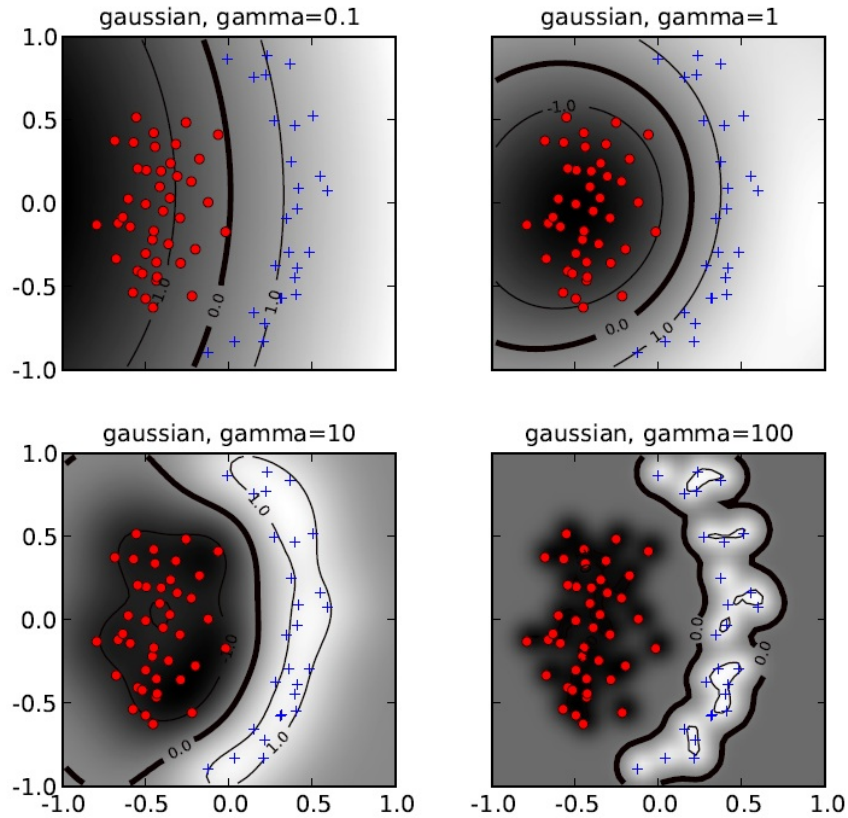


Figura 12: Efeito do parâmetro γ para um valor fixo da constante de penalidade C . para valores baixo de γ o limite da decisão é quase linear. Se o valor de γ aumenta, a flexibilidade do limite de decisão aumenta, conforme o estilo da figura 11.

Fonte: (BEN-HUR; WESTON, 2008)

3 TRABALHOS CORRELATOS

Neste capítulo serão apresentados trabalhos relacionados á utilização do SVM, e também ao método proposto neste trabalho. “... Como um tópico ativo de pesquisas em visão computacional, o monitoramento visual, tem como metas detectar, reconhecer e rastrear objetos em cenas dinâmicas, buscando compreender o comportamento humano” (RÄTY, 2010, p. 497).

Wang et al. (2014) apresentou uma abordagem de transferência automática de detecção genérica de pedestres, para uma detecção de cena específica em vigilância estática com vídeos sem rotulação manual de amostras da cena alvo. O desenvolvimento desta abordagem tem sua justificativa no fato de que a performance da detecção genérica de pedestres cai drasticamente quando aplicada a cenas específicas, devido á diferença entre as fontes usadas no treinamento e as amostras da cena alvo.

Esta abordagem propõe um *framework* de transferência de aprendizado e a exploração de um conjunto de características de contexto para seleção automática de amostras de treinamento. Este processo é composto de três módulos: detecção e seleção de amostras, rotulação de amostras alvo e estimativas de níveis iniciais de confiança, treinamento de detectores de cenas específicas. O processo inicia com um detector genérico de pedestres, aplicado sobre amostras não rotuladas de vídeos coletados das cenas alvo (WANG et al., 2014).

Baseado nos resultados e nas características de contexto, amostras positivas e negativas são selecionadas. Como os rótulos das amostras selecionadas são previstas por pontuações de detecção e características de contexto e podem estar erradas, sua pontuação de detecção é estimada. As amostras selecionadas e sua pontuação de confiança são usadas para treinar o detector de cenas específicas através do *framework*. Então o detector atualizado é aplicado novamente sobre as amostras da cena alvo, para selecionar mais amostras para um novo treinamento. Resultados preliminares foram publicados em (WANG et al., 2012).

Na figura 13 a comparação de resultados do detector de cenas específicas e outras abordagens, (a) Positivas (1^o linha) Negativas (2^o linha) amostras da base de dados *INRIA*. (b) Resultados de detecção na base de dados de tráfego *MIT* e (c) base de dados *CUHK Square*. Na figura á esquerda o resultado da detecção com o detector genérico *HOG-SVM* (*Histogram of Oriented Gradient -SVM*) (DALAL; TRIGGS, 2005) treinado na base de dados *INRIA*. Enquanto na figura á direita o resultado através do Detector de Cenas Específicas (também *HOG-SVM*) (DALAL; TRIGGS, 2005) treinado automaticamente pela abordagem proposta.

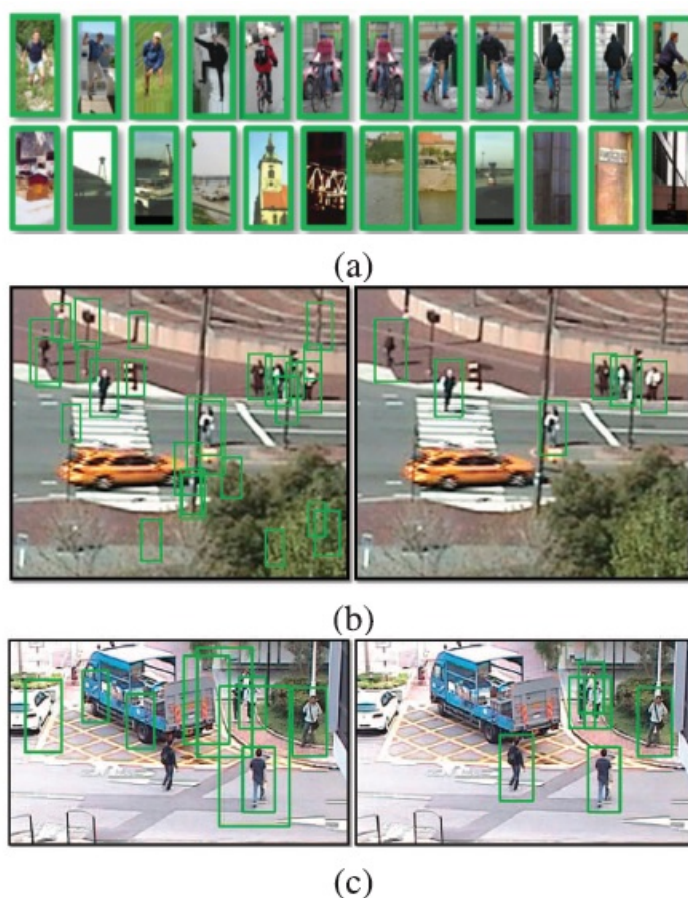


Figura 13: Exemplos de Amostras Positivas e Negativas e comparação entre abordagens
Fonte: (WANG et al., 2014)

As bases de dados *MIT Traffic* e *CUHK* possuem 420 e 350 *frames* na base de treinamento. Todos rotulados para treinar o detector de cena específica, (e) e (f) comparam abordagem de transferência de aprendizado que requer amostras fonte e amostras alvo rotuladas para treinamento. A base de dados *INRIA* e 50 *frames* manualmente rotulados da cena alvo são usados por eles.

A otimização dos parâmetros se deu pela equação 5, e $\mu = \gamma = 1$. Esta abordagem é estável quando μ e γ variam em uma faixa relativamente ampla.

$$C = 1 / \frac{1}{(n_s + n_t (\sum_{i=1}^{n_s} \|x_i^s\| + \sum_{i=1}^{n_t} \|x_i^t\|))^2} \quad (5)$$

Tabela 2: Comparação Final Método *Scene Specific* em relação á outros métodos

Método	MIT Traffic			CUHK Square		
	iterações	Tempo	ção	iterações	Tempo	detecção
Cena Específica	2	4 horas	69,00%	1	1,2	51,00%
Wang CVPR' 11	10	18 horas	62,00%	7	8	44,00%
Nair CVPR' 04	5	2,5 horas	42,00%	4	2	28,00%
Genérico	Não Disp.	Não Disp.	21,00%	Não Disp.	Não Disp.	15,00%
Manual + INRIA	2	1,7 horas	81,50%	2	0,9	55,00%

Fonte: (WANG et al., 2014)

Rahimi et al. (2013) apresentou um método usando três câmeras para rastreamento de pessoas "obstruídas", baseado num *Framework* de filtro de Partículas. Subtração de fundo, *HOG* e, posteriormente o *SVM* são usados para detectar pessoas no primeiro *frame* de cada câmera. Cores e informações *HOG* são usados para realizar a descrição destas, um *framework* de filtro de partículas é usado para rastrear pessoas em cada câmera separadamente.

O monitoramento com apenas uma câmera é pouco eficiente para rastreamento de pessoas "obstruídas", ou seja quando um objeto de interesse está obstruído por outro objeto. O rastreamento multi câmera, que devido ao aumento nos requerimentos dos sistemas de vigilância tem recebido grande atenção, é proposto na tentativa de resolver este tipo de problema. No trabalho proposto quando isto ocorre e o humano está obstruído por outra pessoa ou objeto ele é nomeado como "Humano Obstruído" e a câmera é nomeada como "Visão obstruída"(RAHIMI et al., 2013).

Então esta imagem é dividida em três partes (cabeça, tronco e pernas). No primeiro *frame*, as características *HOG* são consideradas separadamente como um modelo não obstruído, conforme ilustrado na figura 14. Estas características são então comparadas com a mesma parte, do modelo não obstruído através das equações:

$$d_{Head} = \sqrt{1 - \rho} [\hat{q}_{Head}, \hat{p}_{Head}],$$

$$d_{Trunk} = \sqrt{1 - \rho} [\hat{q}_{Trunk}, \hat{p}_{Trunk}],$$

$$d_{Leg} = \sqrt{1 - \rho} [\hat{q}_{Leg}, \hat{p}_{Leg}]$$

Desta forma a obstrução é detectada se: $d_{part} > th_{part}$, e deve se citar que o método proposto tem bom desempenho na presença de obstruções .



Figura 14: Divisão da Região em Três Partes em uma Imagem Obstruída

Fonte: (RAHIMI et al., 2013)

Os resultados em testes do método proposto e o método proposto por (NUMMIARO et al., 2002) e (CHEN et al., 2009) são apresentados na figura: 15

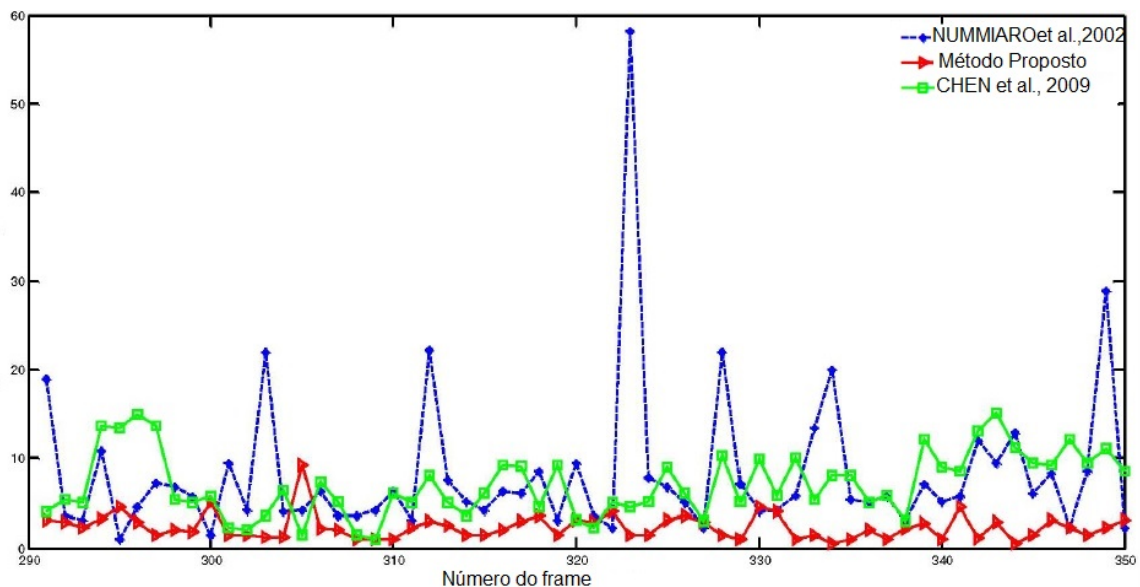


Figura 15: Média de Erros Método Rede Com 3 Câmeras, a linha vermelha mostra a média de erros do método proposto em relação aos demais.

Fonte: (RAHIMI et al., 2013)

Ouanane et al. (2012) apresenta um novo paradigma para detecção de comportamentos violentos (cenas de boxe). Composto por dois níveis: uma análise de baixo nível caracteriza cada *frame* usando um novo descritor geométrico, baseado num estudo de (LACQUANITI; SOECHTING, 1982), que prova que a trajetória dos braços e

pulsos independem da velocidade dos movimentos. A distância Euclidiana d_i é calculada pela fórmula

$$d_i = \sqrt{(x_c - x_i)^2 + (y_c - y)^2} \quad (6)$$

Posteriormente através de uma abordagem baseada em *BOF* (*Bag of Features*) os movimentos locais são detectados. A técnica *BOF* representa os vídeos em conjuntos desordenados de características de espaço tempo, que são quantificadas em vocabulários discretos ou em um livro de códigos. Estas características são associadas ao vizinho mais próximo no livro de códigos (OUANANE et al., 2012).

A análise de alto nível consiste em gerar os vetores de características de uma sequência de vídeo usando um conjunto de rótulos. Então estas cenas de boxe são reconhecidas usando um classificador *SVM*. O classificador *SVM* é responsável por separar as ações em duas classes: agressiva e não agressiva, o *SVM* foi escolhido por ter uma melhor performance de generalização, sendo usado com diferentes *kernels*, especialmente *RBF* e *MLP* (*Multy Layer Perception*).

A estrutura deste processo é ilustrado na figura 16 .

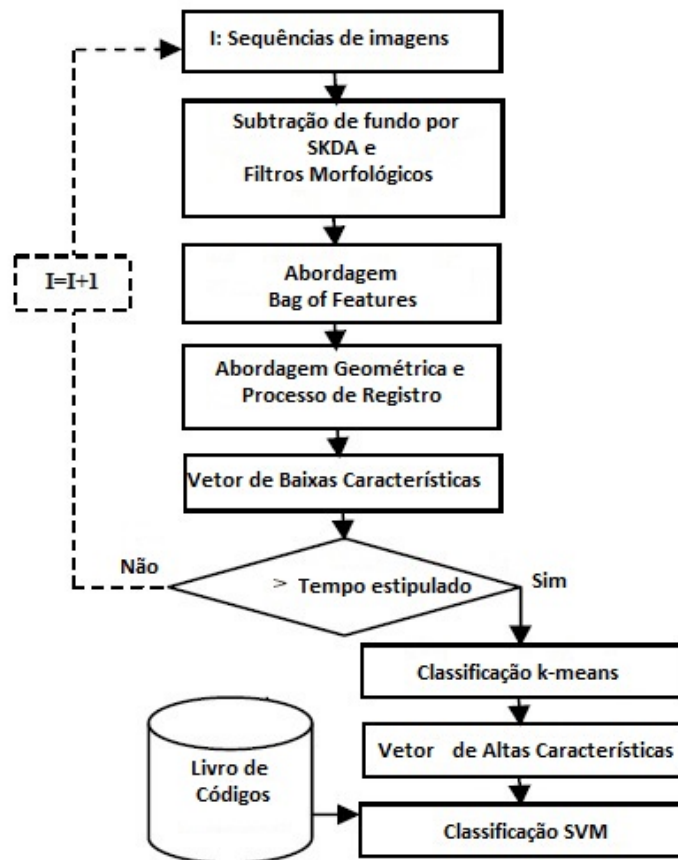


Figura 16: Fluxograma Algoritmo Comportamento Humano Agressivo

Fonte: (OUANANE et al., 2012)

O desempenho do algoritmo foi avaliado utilizando a base de dados *KTH* de ações composto por seis tipos de ações: caminhar, *cooper* (caminhada rápida), correr, bo-xear, *hand-waving* (gestos com as mãos), bater palmas. Cada ação é realizada por 25 pessoas em 4 diferentes cenários: variação de luminosidade, cenários estáticos, diferentes roupas . Desta forma havendo múltiplos ciclos para a mesma ação. E uma base de dados própria que contém cenas de boxe em diferentes cenários.

A 3 mostra os resultados de acurácia para a base de dados própria (boxe).

Tabela 3: Acurácia(base de dados própria, boxe, com *BOF*).

Algoritmos de Classificação			
	SVM-Linear	SVM- RBF	SVM-MLP
Precisão	90.0%	91%	89.4%
Retorno	89.6 %	91%	89.1%

Fonte: (OUANANE et al., 2012)

A tabela 4 mostra os resultados de acurácia para a base de dados *KTH* (boxe).

Tabela 4: Acurácia(base de dados *KTH*, boxe, com *BOF*).

Algoritmos de Classificação			
	SVM-Linear	SVM- RBF	SVM-MLP
Precisão	87.9%	88.5%	87.1%
Retorno	87.5%	89.1%	87.9%

Fonte: (OUANANE et al., 2012)

Nota-se que a média de reconhecimento chega a 90% para a base de dados própria e 87% para a base de dados *KTH*.

Agora a abordagem geométrica é associada ao *BOF*, e os resultados apresentados na tabela 5 .

Tabela 5: Acurácia(base de dados própria, boxe, com Abordagem Geométrica).

Algoritmos de Classificação			
	SVM-Linear	SVM- RBF	SVM-MLP
Precisão	97.0%	98.2%	97.4%
Retorno	97.0%	98.2%	97.6%

Fonte: (OUANANE et al., 2012)

Na tabela 6 os resultados da base *KTH* com abordagem geométrica associada ao *BOF*.

Tabela 6: Acurácia(base de dados KTH, boxe, com Abordagem Geométrica

	Algoritmos de Classificação		
	SVM-Linear	SVM- RBF	SVM-MLP
Precisão	95.0%	97.0%	96.8%
Retorno	95.1%	97.0%	96.1%

Fonte: (OUANANE et al., 2012)

A tabela 7 apresenta os resultados comparando diferentes métodos para a base de dados *KTH*.

Tabela 7: Comparação de taxas de reconhecimento entre diferentes métodos base de dados KTH

Método	Taxa de Reconhecimento	Classificador
Método Proposto	96.00	SVM
Liu 2008	94.20	2-NN
Niebles 2007	72.00	L-SVM
Dollar 2005	81.17	3-NN
Kim 2009	95.33	3-NN

Fonte: (OUANANE et al., 2012)

O método proposto proporciona redução significativa de vetores de características de dimensão e evita alto custo computacional. O método também é mais robusto quando há mudanças na luminosidade e tem um bom desempenho quando há oclusão causada pelo elemento causador do movimento.

Ouanane et al. (2013) apresenta um algoritmo de decisão para detecção de comportamento humano. Utilizando *SURF (Speeded Up Robust Features)* que proporciona um alto nível de características de espaço tempo, usadas como características invariáveis para melhorar o reconhecimento do comportamento humano, e técnicas *PCA (Principal Component Analysis algorithm)*.

Esta robustez do *SURF* será usada para obter as características espaço tempo que modelarão a variação de comportamentos agressivos ou não agressivos. O algoritmo *PCA* é usado para reduzir a dimensionalidade das características adquiridas e fornecer um padrão robusto.

Posteriormente o *PCA* é usado como uma entrada para o *SVM*, que será capaz de classificar os comportamentos em agressivo e não agressivo. A escolha do *SVM* é justificada pela sua eficácia em espaços de alta dimensão, e por usar subconjuntos de pontos de treinamento na função de decisão (Vetores de Suporte), que o torna possivelmente mais ágil que outros classificadores. O *SVM* procura um plano de decisão com a maior distância do modelo de treinamento dado:

$$\{(x_i, y_i) \mid y_i = 1 \text{ or } -1, \quad i = 1, \dots, N\} \quad (7)$$

onde x_i é o espaço de entrada, y_i é o rótulo da classe e N é o número de amostras de treinamento.

Para validar o desempenho do algoritmo, testes foram executados com a base de dados *KTH* de ações, composto por seis tipos de ações: caminhar, *cooper* (caminhada rápida), correr, boxear, *hand-waving* (gestos com as mãos) e bater palmas. Destas ações, o boxe, gestos com as mãos e bater palmas podem ser consideradas comportamento agressivo.

O classificador *SVM* é então aplicado utilizando diferentes *kernels* entre eles, *Polynomial*, e *RBF* para melhorar a taxa de acurácia. Na tabela 8 pode se observar que a melhor taxa de acurácia foi obtida com o *kernel RBF*, com acurácia de 96.80%. O método também foi submetido a testes comparativos com o método apresentado por (CHEN et al., 2008).

Tabela 8: Média Taxa de Acurácia

Média Taxa de Acurácia	
SVM-Linear	92.00%
SVM-Quadratic	95.00%
SVM- Polynomial	95.00%
SVM- RBF	96.80%

Fonte: (OUANANE et al., 2013)

A tabela 9 mostra as taxas de acurácia comparando o método proposto com as técnicas em evidência no momento. Podemos observar que o método proporciona desempenho satisfatório e é eficaz. Pode se observar também que a taxa de acurácia se sobrepõe aos demais métodos.

Tabela 9: Média Taxa de Acurácia comparando com Estado da arte

Métodos	Taxa de Acurácia
Método Proposto	96.00%
Wang et al	94.20%
Schindler et al	92.70%
Laptev et al	91.80%

Fonte: (OUANANE et al., 2013)

4 MÉTODO PROPOSTO

O método adotado para o desenvolvimento deste trabalho foi constituído de duas partes.

A primeira parte envolveu testes executados manualmente (tentativas sucessivas) para definir valores para os parâmetros *cost* e *gamma*, que tem influência direta no desempenho do algoritmo *SVM*. A segunda parte fez uso do recurso da ferramenta *Weka* chamado *CVParameterSelection*, que faz uma busca automatizada pelos valores ideais para os parâmetros (*Grid-Search*).

4.1 METODOLOGIA DOS TESTES

Inicialmente foram realizados testes sobre as bases de dados com os os parâmetros *defaults*, ou seja, os valores definidos como padrão nas configurações do software *Weka* (*Kernel = RBF*, $C = 1$ e $\gamma = 0,0$). Considerando-se que os algoritmos de classificação, generalizando, procuram aprender, as fases iniciais são de suma importância. Desde as bases de dados usadas para o treinamento, *kernel* escolhido e a parametrização.

O algoritmo foi executado utilizando uma variação da técnica de *Cross-validation*, o *Multifold-cross-validation* que divide a base de dados contendo N instâncias em K subconjuntos, como no estudo em questão, em 10 partes e utiliza $K-1$ para treinamento e o restante para testes.

Este processo é executado K vezes, sucessivamente. Isto faz com que haja um custo alto em termos de tempo, dependendo do número de instâncias. Em seguida também usada a técnica de *percentage-split*. Assim foram obtidos os primeiros resultados e estes resultados foram comparados com os obtidos em estudos anteriores, com as mesmas bases de dados e com os mesmos recursos. Tiveram início então os testes onde os valores para os parâmetros foram atribuídos de forma não automati-

zada.

A figura 17 mostra um exemplo da aplicação do *SVM* e a inserção dos valores para os parâmetros nas configurações do *Weka*.

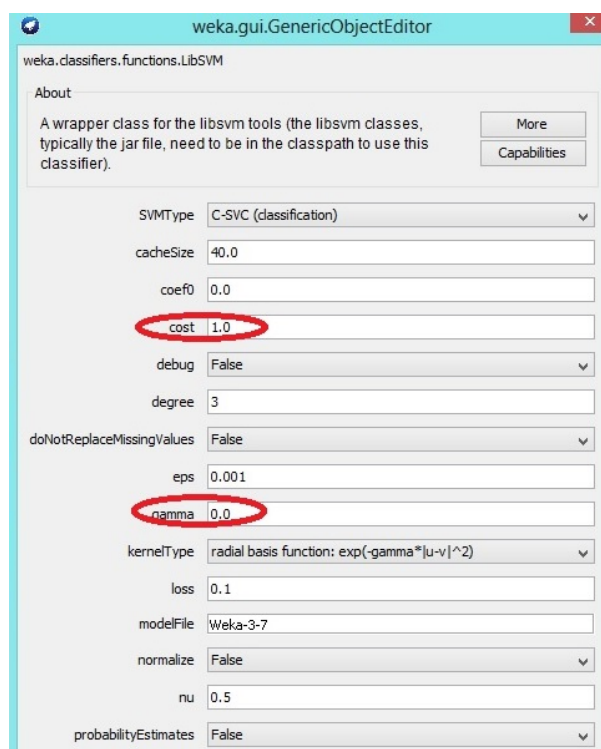


Figura 17: Exemplo de tela de teste não automatizado de parâmetros

Fonte: Software Weka

4.1.1 Ambiente *Weka*

O *Weka* é uma ferramenta *open source*, desenvolvida pelo grupo de Aprendizado de máquina da Universidade de *Waikato*. Este *software* fornece um conjunto de algoritmos de *ML* de ampla aplicação.

(MALLIOS et al., 2011) aponta o *Weka* como sendo “... um poderoso ambiente para numerosas atividades de *Data Mining*”. Implementa vários tipos de técnicas de classificação, algoritmos para regressão e ferramentas de visualização.

O ambiente *Weka* tem sido amplamente utilizado na área de pesquisa de *ML*, um estudo recente utilizando o *CVPParameterSelection* na área de medicina foi realizado por (YOO et al., 2013, p. 456).

4.1.2 Bases de Vídeos Utilizadas

A base de dados *UT-Interaction dataset* (apenas a primeira parte), composta de 10 sequencias de vídeos foi a base de dados utilizada para a execução deste trabalho, esta base de dados contém vídeos de execuções de 6 classes de interações humano-humano: aperto de mão, apontar, abraçar, empurrar, chutar e socar.

Esta base de dados foi o apresentada no evento SDHA2010 1 - ICPR 2010 *Contest on Semantic Description of Human Activities*. Este evento foi realizado com o intuito de incentivar pesquisas na área do reconhecimento de atividades humanas, que poderão ser aplicadas, por exemplo, em sistemas de vigilância. Cada vídeo contém diversas interações entre humanos (por exemplo, apertos de mão) que ocorrem em sequencia e/ou simultaneamente.

No trabalho apresentado por (RYOO; AGGARWAL, 2009) encontram-se exemplos de utilização desta base. Várias pessoas com mais de 15 combinações diferentes de roupas aparecem nos vídeos. Os vídeos são gravados com a resolução de 720 X 480 *pixels*, 30 quadros por segundo, sendo que a altura de uma pessoa no vídeos é de cerca de 200 *pixels*.

As classes de interação (apertar as mãos), está exemplificada na Figura 18.

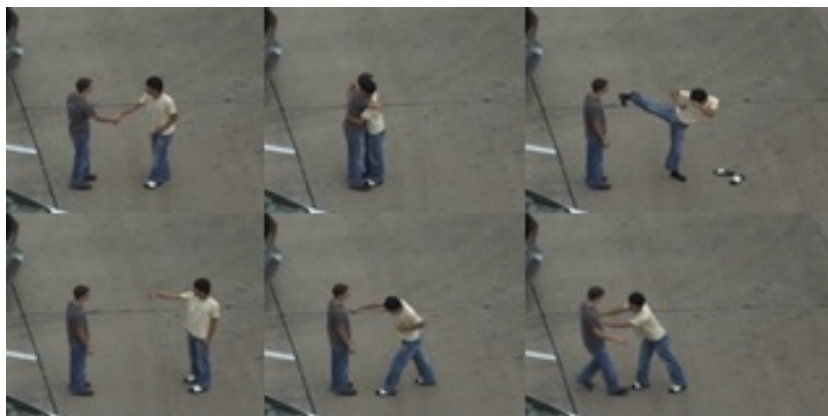


Figura 18: Ações - base de vídeos ICPR. Aperto de mãos

Fonte: (RYOO; AGGARWAL, 2009)

4.1.3 Testes Não Automatizados

Nesta etapa foram executados primeiramente os testes utilizando o *kernel RBF*, que até o momento seria o mais promissor. Conforme já citado, os testes foram manuais (o que seria um *grid-search*, porém de forma não automatizada), para cada arquivo da base de dados eram inseridos manualmente os valores para os parâmetros C e γ . Valores estes variando a cada 0,001 (valores lineares).

Após várias tentativas, e obviamente até um valor determinado (um valor máximo foi tentado), chegou-se a um valor o qual, mesmo ultrapassando este valor, não houve mudanças significativas nos resultados das classificações. Estes valores foram: $C = 1$, $\gamma = 0.001$.

Posteriormente os mesmos testes foram executados envolvendo o *kernel Linear*, que por sua vez também proporcionou bons resultados. Este *kernel* especificamente foi o que demonstrou maior custo em termos de tempo.

4.1.4 Resultados de Testes Não Automatizados

A tabela 10 mostra a porcentagem positiva de acertos do *RBF* sobre o *Linear* em cada uma das sequencias da base de dados. Cada uma destas sequências possui um valor diferente de instâncias.

O *kernel RBF* proporcionou o melhor resultado, sendo que na sequencia07.avi obteve resultado 28,61% acima sobre o *kernel Linear*. Na sequencia 04.avi, obteve 97,39% de acertos.

Tabela 10: Porcentagens do melhor *kernel*—*kernel RBF*

	% Corretas	% Incorretas	% <i>RBF</i> Sobre <i>Linear</i>
Seq01.avi <i>RBF</i> Manual	92,70	7,30	10,31%
Seq01.avi <i>Linear</i> Manual	82,39	17,61	
Seq02.avi <i>RBF</i> Manual	96,22	3,78	1,41%
Seq02.avi <i>Linear</i> Manual	94,81	5,19	
Seq03.avi <i>RBF</i> Manual	93,06	6,94	6,56%
Seq03.avi <i>Linear</i> Manual	86,50	13,50	
Seq04.avi <i>RBF</i> Manual	97,39	2,61	2,61%
Seq04.avi <i>Linear</i> Manual	94,78	5,22	
Seq05.avi <i>RBF</i> Manual	91,06	9,94	13,75%
Seq05.avi <i>Linear</i> Manual	77,31	22,69	
Seq06.avi <i>RBF</i> Manual	93,13	6,87	8,64%
Seq06.avi <i>Linear</i> Manual	84,49	15,51	
Seq07.avi <i>RBF</i> Manual	92,34	7,66	28,61%
Seq07.avi <i>Linear</i> Manual	63,73	36,26	
Seq08.avi <i>RBF</i> Manual	92,66	7,34	6,53%
Seq08.avi <i>Linear</i> Manual	86,13	13,86	
Seq09.avi <i>RBF</i> Manual	96,13	3,87	5,11%
Seq09.avi <i>Linear</i> Manual	91,02	8,98	
Seq10.avi <i>RBF</i> Manual	97,19	2,81	10,07%
Seq10.avi <i>Linear</i> Manual	87,12	12,87	

Na figura 19 é apresentado o resultado desta etapa em forma de gráfico.

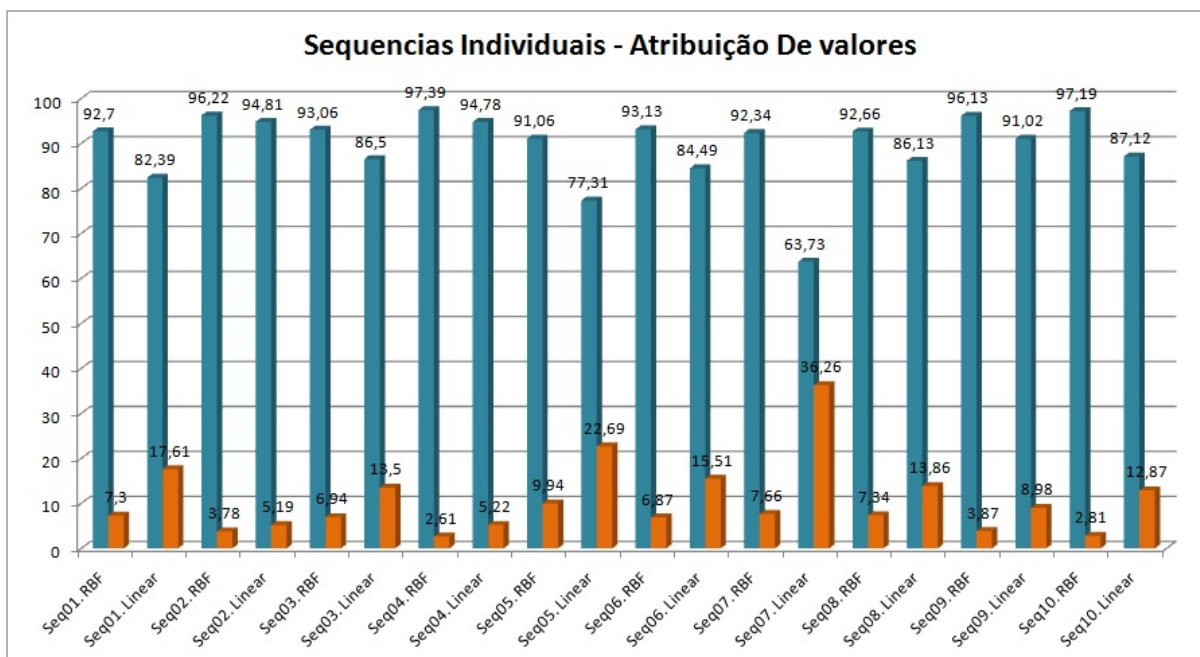


Figura 19: Gráfico comparativo entre testes Não Automatizados envolvendo *kernel RBF* e *kernel Linear*.

Fonte: Autoria própria

4.1.5 Análise dos resultados dos Testes Não Automatizados

Estes testes tiveram um alto custo em termos de tempo, pois a cada iteração era preciso alterar os valores dos parâmetros manualmente, e manter um controle dos valores já testados. Pode se observar claramente na tabela 11 que o desempenho nesta fase, de testes não automatizados, foi melhor utilizando o *kernel RBF*. Este *kernel* obteve um desempenho em média de 9,36 % acima, sobre o *Linear*.

O *kernel linear* teve um desempenho abaixo do *RBF*, porém demonstrou ser capaz de realizar boas classificações. O que o torna uma opção a ser considerada adiante nos testes.

Tabela 11: Comparação Final Testes Não Automatizados

	Total de Instâncias	Corretas	Incorretas	% Correta
RBF Manual	25648	24462	1186	95,38%
Linear Manual	25648	22239	3409	86,71%

4.2 TESTES AUTOMATIZADOS

4.2.1 Alteração nos atributos usados

Após testes preliminares foi constatado que usando diferentes atributos (características), os percentuais de acertos foram maiores. Desta forma novos atributos (características) foram escolhidos para utilização nos testes, este procedimento é indicado por (WU et al., 2013, p. 214).

A razão que explica o melhor desempenho utilizando estes novos atributos se deve ao fato de que, tais atributos são usados diretamente na descoberta dos movimentos, isto pela aplicação que gerou as bases de dados utilizadas nos testes. Desta forma proporcionando melhores resultados

4.2.2 Grid-Search Bases de Dados Individuais com valores lineares

Neste experimento foi usado um recurso semi automatizado de meta aprendizado, o *CVParameterSelection* da ferramenta *Weka*. O mesmo recurso foi usado por (YOO et al., 2013) obtendo bons resultados com bom desempenho da ferramenta.

Através dele é possível especificar o *range* (faixa) de parâmetros, os valores iniciais e finais, para que a ferramenta execute busca e testes com a finalidade de encontrar os valores ideais para os parâmetros *cost* e *gamma*, de forma que o algoritmo obtenha um resultado com o melhor desempenho.

Nesta fase os testes demoraram cerca de 5 minutos os que executaram em menor tempo, isto dependendo da quantidade de instâncias, e alguns que demoraram mais de 280 horas, que foi o caso da sequencia09.avi, que demorou 12 dias. Os valores escolhidos foram: $C = 0.001$ até 3.0 e $\gamma = 0.001$ até 3.0. O valor 0.001 foi escolhido por ter apresentado bons resultados na fase anterior.

A figura 20, ilustra esta funcionalidade do *Weka*.

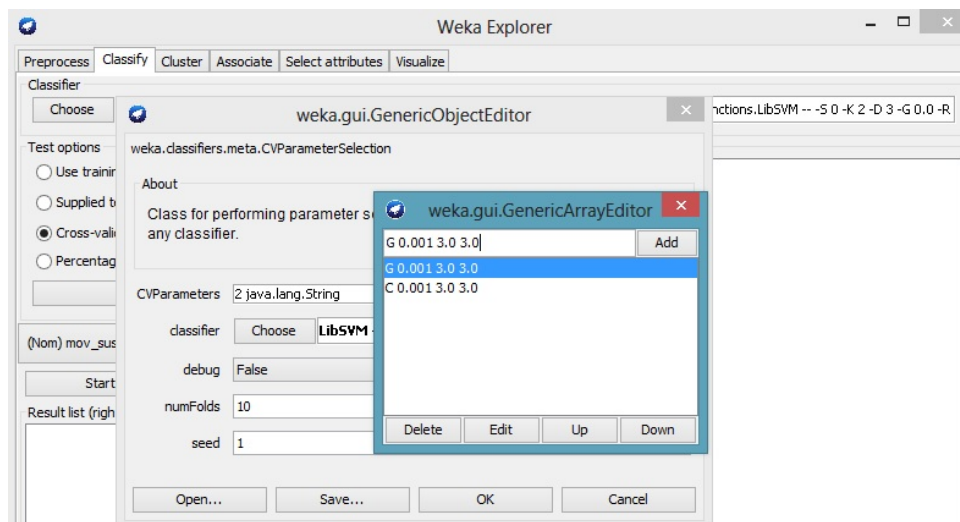


Figura 20: Exemplo do uso do CVParameterselection.

Fonte: Software Weka

4.2.3 Análise dos Resultados Grid-Search Bases de Dados Individuais, com valores lineares

O *kernel RBF* mantém um desempenho acima do *kernel Linear*. Este desempenho obteve uma porcentagem ligeiramente acima usando a técnica de *multifold-cross-validation* sobre a técnica de *percentage-split*. Esta diferença foi de 0,52 %.

Isot pode ser observado na tabela 12, onde os resultados das sequências de 1 a sequência 9 da base de dados estão somadas.

Já o *kernel linear*, utilizado em conjunto com a técnica de *percentage-split*, obteve 0,53 % acima, sobre a técnica de *multifold-cross-validation*. Comparando o melhor desempenho do *kernel RBF* e o melhor desempenho do *kernel linear*, a melhor média de acertos foi do *RBF*, com 6,91 % acima.

Tabela 12: Comparação Final Bases de Dados Individuais

	Nº Instâncias	Corretas	Incorretas	% Correta
RBF Folds	25648	24620	1028	95,99%
RBF Split	8720	8325	395	95,47%
Linear Folds	25648	22713	2935	88,56%
Linear Split	8720	7770	950	89,11%

4.3 GRID-SEARCH COM VALORES EXPONENCIAIS

Para esta etapa de testes foram usados valores exponenciais, que é o que recomenda (HSU et al., 2010) e os valores ficaram desta forma : ($C = 2^{-5}, 2^{-3}, \dots, 2^{15}$) e ($\gamma = 2^{-9}, 2^{-7}, \dots, 2^3$). Lembrando que o parâmetro γ não se aplica ao *kernel linear*.

Na tabela 13 nota se que o *kernel RBF* mais uma vez obteve resultados superiores, com uma porcentagem ligeiramente acima usando a técnica de *multifold-cross-validation* sobre técnica de *percentage-split*. Esta diferença foi de 0,44%.

O *kernel Linear* manteve sua média de desempenho, em conjunto com a técnica de *percentage-split* obteve um desempenho de 0,32% sobre o *multifold-cross-validation*. A comparação entre o melhor desempenho do *kernel RBF* e do *kernel Linear*, aponta para uma melhor média de acertos para o *kernel RBF* na ordem de 7,11%.

Tabela 13: Comparação Final Com Valores Exponenciais

	Nº Instâncias	Corretas	Incorretas	% Correta
RBF_Folds	25648	24642	1006	96,08%
RBF_Split	8720	8340	380	95,64%
Linear_Folds	25648	22736	2882	88,65%
Linear_Split	8720	7758	962	88,97%

4.4 TESTES COM BASE ÚNICA

Nesta etapa as bases de dados foram unidas, a fim de verificar se o desempenho do algoritmo com uma base de dados com um número maior de instâncias e os resultados obtidos conforme a tabela 14.

Tabela 14: Testes Com Base Única

	NºInstâncias	(+)	% (+)	(-)	% (-)	Parâmetros
unica_RBF_Folds	25648	24597	95.90	1051	4.10	$\gamma = 10^6, C = 10^{10}$
unica_RBF_Split	8720	8290	95.06	430	4.96	$\gamma = 10^6, C = 10^{10}$
unica_Linear_Folds	25648	21529	83,94	4119	16,06	$C = 10^3$
unica_Linear_Split	8720	7443	85,35	1277	14,65	$C = 10^3$
unica_Polynomial_Folds	25648	18409	71,77	7239	28,23	$\gamma = 10^3, C = 10^6$
unica_Polynomial_Split	8720	6232	71,46	2488	28,54	$\gamma = 10^3, C = 10^6$

Na figura 21 é apresentado o resultado desta etapa em forma de gráfico.

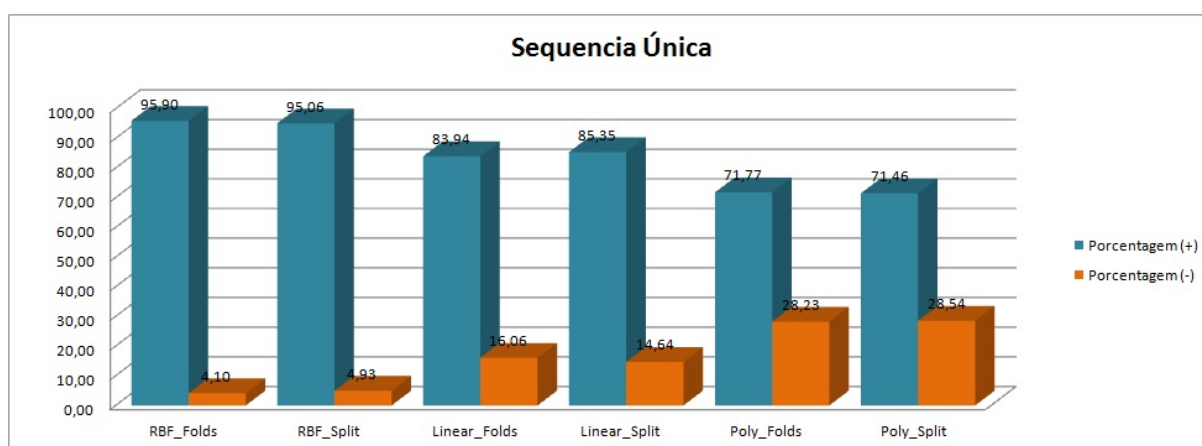


Figura 21: Gráfico Comparativo do Teste com Sequencia Única.

Fonte: Autoria própria

4.5 RESULTADO FINAL

O objetivo principal deste trabalho é alcançar as melhores taxas de acertos na detecção de comportamentos suspeitos em cenas de vídeos. A tabela 15 mostra o resultado obtido em um estudo anterior, onde foi obtida uma taxa de acertos de 86,40%, utilizando a técnica de *multifold-cross-validation* em conjunto com o *kernel RBF*.

Este estudo não utilizou otimização de parâmetros. Este foi o referencial de partida para iniciar este trabalho. Na tabela compara-se este estudo com o método proposto neste trabalho, para esta comparação foi escolhida a melhor taxa de acertos obtida, neste caso a taxa de foi de 95,90%.

Tabela 15: Resultado Inicial x Resultado Final

	Nº Instâncias	(+)	% (+)	(-)	% (-)
Seq.unica_RBF_Folds_Final	25648	24642	96,08%	1006	3,02%
Seq.unica_RBF_Folds_Inicial	25978	22445	86,40%	3533	13,60%

Fonte: Autoria própria

Para chegar-se a este resultado alguns testes foram realizados conforme já citado anteriormente, e este resultado foi obtido utilizando-se da técnica de *multifold-cross-validation* em conjunto com o *kernel RBF*.

A porcentagem de acertos foi elevada em 9,50% em relação ao estudo anterior. Entretanto, pode-se observar que utilizando as bases de dados individualmente o desempenho teve uma pequena diferença conforme ilustrado na tabela 16, que foi de 0,18% e a porcentagem de acertos neste caso elevada em 9,68%.

Tabela 16: Resultados em Sequências Individuais

	Nº Instâncias	(+)	% (+)	% (-)	% (-)	Parâmetros
unica_RBF_Folds	25648	24642	96,08%	1006	3,92%	$\gamma = 10^6, C = 10^{10}$
unica_RBF_Split	8720	8340	95,64%	380	4,36%	$\gamma = 10^6, C = 10^{10}$
unica_Linear_Folds	25648	22736	88,65%	2882	11,24%	$C = 10^3$
unica_Linear_Split	8720	7758	88,97%	962	11,03%	$C = 10^3$

Fonte: Autoria própria

4.6 DISCUSSÕES FINAIS

Observou-se nos testes que:

- Os mesmos valores de parâmetros proporcionam diferentes resultados quando aplicados em diferentes tipos de *kernel*.
- No *grid-search*, há ao longo da faixa de valores testados, alguns valores "correspondentes", ou seja, que podem produzir um mesmo resultado de classificação e de acurácia. Algo semelhante foi detectado no trabalho de (GAO et al., 2013).

- O *kernel RBF* em geral tem um melhor desempenho que o Linear, mas pode variar dependendo do tipo da base de dados.
- O *kernel Linear* por ter apenas um parâmetro a ser testado, é uma opção a ser considerada para testes iniciais em algum trabalho correlato, desta forma reduz a complexidade dos testes visto que o *Kernel RBF* possui dois parâmetros.
- No trabalho apresentado por (NOVAKOVIC; VELJOVIC, 2011), os estudos mostraram que sem otimização de parâmetros, a precisão na classificação foi melhor utilizando o *kernel linear*, e que após a otimização dos parâmetros a precisão na classificação foi melhor para todos os *kernels*, não havendo até mesmo uma grande variação nos resultados. O estudo em questão trata de classificação em bases de dados clínicas.

No método proposto neste trabalho, observou-se que esta informação não é válida para todas as situações. Os resultados alcançados neste trabalho mostram que mesmo sem a parametrização, o *kernel RBF* proporcionou melhores resultados.

5 CONCLUSÃO E TRABALHOS FUTUROS

5.1 CONCLUSÃO

Para realização deste trabalho foi definido que seria utilizado primeiramente o método de atribuição de valores para otimização de parâmetros. E após este, seria utilizado o *grid-search*. Por este motivo mesmo após ter sido alcançado um resultado próximo ao esperado os experimentos continuaram, para que fosse possível a comparação entre os dois métodos.

Após realizar os experimentos, constatou-se que:

O método de atribuição de valores é eficiente na obtenção de taxas relevantes no reconhecimento de comportamentos suspeitos. Como desvantagem este método pode ser vulnerável a falhas, no sentido de que, como o método é manual, está suscetível ao erro humano. Outra desvantagem é o alto custo de tempo para realização dos experimentos.

Entre as ferramentas pesquisadas para otimização de parâmetros, a ferramenta Weka foi a de mais facilidade e praticidade de utilização, em especial o recurso *CVPParameterSelection* que permite a execução automatizada da otimização de parâmetros. Por este motivo foi confirmada como a melhor opção para otimização de parâmetros.

O *grid-search* automatizado proporcionou resultado superior em relação à atribuição de valores, quando aplicado com o *range* de parâmetros que, durante os experimentos com atribuição de valores demonstraram o melhor resultado, foi em pequena porcentagem superior. Porém já houve ganho em termos de tempo altamente satisfatório, devido ao fato de que o *range* de parâmetros é inserido nas configurações do *Weka*, e testado automaticamente.

Esta mesma técnica aplicada então, com valores exponenciais, proporcionou o melhor resultado final. O valor obtido foi de 96,08% de acertos contra 86,40% sem a otimização de parâmetros.

Quanto aos tipos de *kernel*, o que obteve desempenho superior foi o *kernel RBF*, isto ocorreu em todas as etapas de experimentos, tanto na atribuição de parâmetros quanto nos testes automatizados. Sendo assim confirmado como a melhor opção no reconhecimento de movimentos suspeitos. Na literatura pesquisada, este *kernel* já era apontado como uma opção promissora em diversas áreas de pesquisa.

Os resultados obtidos provam que, o algoritmo *SVM*, que tem obtido um desempenho notável no contexto de *ML* em diversas ramificações, é capaz de proporcionar excelentes resultados também no reconhecimento de movimentos suspeitos em cenas de vídeo, além das aplicações citadas neste trabalho.

Quanto ao método proposto, adicionou rapidez no processo de otimização de parâmetros, pois o *CVParameterSelection* executa os testes sendo apenas necessário especificar o *range* de parâmetros. O ganho em termos de tempo é incomparável. Por ser automatizado o método é seguro, pois não é suscetível ao erro humano como esquecimento, suposições ou desatenção.

Destaca-se a relevante influência da escolha dos atributos (características) usados ao treinar as bases de dados. Como mencionado neste trabalho, ao alterar-se estes atributos, o resultado foi superior. Assim também destaca (WU et al., 2013, p. 214)“...além de se escolher as melhores características, é importante escolher as características que proporcionem resultado superior”.

5.2 TRABALHOS FUTUROS

Como trabalhos futuros podemos listar:

- Um estudo sobre os movimentos detectados erroneamente e corretamente antes e depois da otimização.
- Uma avaliação de quais movimentos são frequentemente detectados com erro e propor soluções que contornem estes erros.
- Analisar a influência dos tipos de *kernel* no sentido de detectar a cada tipo de *kernel* quais movimentos são classificados com e sem erro.
- Utilização do ambiente *Experimenter* na ferramenta *Weka*.

- Realizar uma avaliação do desvio padrão ocorrido nas taxas de acertos neste trabalho.

REFERÊNCIAS

ASADPOUR MOHAMMD REZA RAVANFAR, R. F.-R. V. 2013. Disponível em: <<http://www.intechopen.com/books/export/citation/BibTex/brain-computer-interface-systems-recent-progress-and-future-prospects/adaptive-network-fuzzy-inference-systems-for-classification-in-a-brain-computer-interface>>.

BAI, J. et al. Parameters optimization and application of v-support vector machine based on particle swarm optimization algorithm. In: **Computing, Measurement, Control and Sensor Network (CMCSN), 2012 International Conference on**. [S.l.: s.n.], 2012. p. 113–116.

BEN-HUR, A.; WESTON, J. A user's guide to support vector machines. v. 609, November 2008. Disponível em: <<http://www.springerprotocols.com/Abstract/doi/10.1007/978-1-60327-241-4-13>>.

BORGES, P.; CONCI, N.; CAVALLARO, A. Video-based human behavior understanding: A survey. **Circuits and Systems for Video Technology, IEEE Transactions on**, v. 23, n. 11, p. 1993–2008, 2013. ISSN 1051-8215.

BOUZEGZA, M.; ELARBI-BOUDHIR, M. Automatic understanding of human behavior in videos: A review. In: **Systems, Signal Processing and their Applications (WoS-SPA), 2013 8th International Workshop on**. [S.l.: s.n.], 2013. p. 185–190.

CHANG, C.-C.; LIN, C.-J. Libsvm: A library for support vector machines. **ACM Trans. Intell. Syst. Technol.**, ACM, New York, NY, USA, v. 2, n. 3, p. 27:1–27:27, maio 2011. ISSN 2157-6904. Disponível em: <<http://doi.acm.org/10.1145/1961189.1961199>>.

CHEN, D. et al. A.: Recognition of aggressive human behavior using binary local motion descriptors. In: **Engineering in Medicine and Biology Society**. [S.l.: s.n.], 2008. p. 5238–5241.

CHEN, R. et al. **Particle-filter-based object tracking with color and texture information fusion**. 2009. 74952F-74952F-8 p. Disponível em: <<http://dx.doi.org/10.1117/12.834212>>.

DALAL, N.; TRIGGS, B. Histograms of oriented gradients for human detection. In: **Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on**. [S.l.: s.n.], 2005. v. 1, p. 886–893 vol. 1. ISSN 1063-6919.

GAO, M.-L. et al. Object tracking using firefly algorithm. **Computer Vision, IET**, v. 7, n. 4, p. 227–237, August 2013. ISSN 1751-9632.

GAO, Y.; SUN, S. An empirical evaluation of linear and nonlinear kernels for text classification using support vector machines. In: **Fuzzy Systems and Knowledge Discovery (FSKD), 2010 Seventh International Conference on**. [S.l.: s.n.], 2010. v. 4, p. 1502–1505.

HSU, C. wei; CHANG, C. chung; LIN, C. jen. **A practical guide to support vector classification**. 2010.

HU, W. et al. A survey on visual surveillance of object motion and behaviors. **Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on**, v. 34, n. 3, p. 334–352, 2004. ISSN 1094-6977.

HUANG, C.-L.; WANG, C. jen. **A GA-based feature selection and parameters optimization for support vector machines**. 2006.

JAIN, A.; DUIN, R.; MAO, J. Statistical pattern recognition: a review. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, v. 22, n. 1, p. 4–37, jan 2000.

JIN, Y. et al. Study on influences of model parameters on the performance of svm. In: **Electrical and Control Engineering (ICECE), 2011 International Conference on**. [S.l.: s.n.], 2011. p. 3667–3670.

KAUTZ, H. **A Formal Theory of Plan Recognition**. Tese (Doutorado) — Dept. of Computer Science, University of Rochester, 1987.

KOU, D.; ZHANG, Y.; ZHENG, H. A parameters selection method of svm. In: **Computational Intelligence and Software Engineering (CiSE), 2010 International Conference on**. [S.l.: s.n.], 2010. p. 1–4.

LACQUANITI, F.; SOECHTING, J. Coordination of arm and wrist motion during a reaching task. **J Neurosci**, v. 2, n. 4, p. 399–408–, abr. 1982. Disponível em: <<http://europepmc.org/abstract/MED/7069463>>.

LEIGHTLEY, D. et al. Human activity recognition for physical rehabilitation. In: **Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on**. [S.l.: s.n.], 2013. p. 261–266.

LIU, S.; JIANG, N. Svm parameters optimization algorithm and its application. In: **Mechatronics and Automation, 2008. ICMA 2008. IEEE International Conference on**. [S.l.: s.n.], 2008. p. 509–513.

MALLIOS, N.; PAPAGEORGIOU, E.; SAMARINAS, M. Comparison of machine learning techniques using the weka environment for prostate cancer therapy plan. In: **Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2011 20th IEEE International Workshops on**. [S.l.: s.n.], 2011. p. 151–155. ISSN 1524-4547.

NOVAKOVIC, J.; VELJOVIC, A. C-support vector classification: Selection of kernel and parameters in medical diagnosis. In: **Intelligent Systems and Informatics (SISY), 2011 IEEE 9th International Symposium on**. [S.l.: s.n.], 2011. p. 465–470.

NUMMIARO, K.; KOLLER-MEIER, E.; GOOL, L. J. V. Object tracking with an adaptive color-based particle filter. In: **Proceedings of the 24th DAGM Symposium on Pattern Recognition**. London, UK, UK: Springer-Verlag, 2002. p. 353–360. ISBN 3-540-44209-X. Disponível em: <<http://dl.acm.org/citation.cfm?id=648287.756370>>.

OUANANE, A.; SERIR, A.; DJELAL, N. Recognition of aggressive human behavior based on surf and svm. In: **Systems, Signal Processing and their Applications (WoSSPA), 2013 8th International Workshop on**. [S.l.: s.n.], 2013. p. 396–400.

OUANANE, A.; SERIR, A.; KEROUH, F. New geometric descriptor for the recognition of aggressive human behavior. In: **Image and Signal Processing (CISP), 2012 5th International Congress on**. [S.l.: s.n.], 2012. p. 148–153.

PRIYANKA, C.; GUPTA, D. Identifying the best feature combination for sentiment analysis of customer reviews. In: **Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference on**. [S.l.: s.n.], 2013. p. 102–108.

RAHIMI, S.; AGHAGOLZADEH, A.; SEYEDARABI, H. Detection and tracking of occluded humans in three-camera network. In: **Robotics and Mechatronics (ICRoM), 2013 First RSI/ISM International Conference on**. [S.l.: s.n.], 2013. p. 32–37.

RAJKUMAR, N.; JAGANATHAN, P. A new rbf kernel based learning method applied to multiclass dermatology diseases classification. In: **Information Communication Technologies (ICT), 2013 IEEE Conference on**. [S.l.: s.n.], 2013. p. 551–556.

RÄTY, T. Survey on contemporary remote surveillance systems for public safety. **Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on**, v. 40, n. 5, p. 493–515, Sept 2010. ISSN 1094-6977.

REN, Y.; BAI, G. Determination of optimal svm parameters by using ga/pso. **JCP**, v. 5, n. 8, p. 1160–1168, 2010. Disponível em: <<http://dblp.uni-trier.de/db/journals/jcp/jcp5.htmlRenB10>>.

RYOO, M. S.; AGGARWAL, J. K. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In: **IEEE International Conference on Computer Vision (ICCV)**. [S.l.: s.n.], 2009.

SCHOLTEN, M. et al. **Optimization of support vector machine (SVM) for object classification**. 2012. 839806-839806-9 p. Disponível em: <<http://dx.doi.org/10.1117/12.923483>>.

SOLIMAN, O.; MAHMOUD, A. A classification system for remote sensing satellite images using support vector machine with non-linear kernel functions. In: **Informatics and Systems (INFOS), 2012 8th International Conference on**. [S.l.: s.n.], 2012. p. BIO–181–BIO–187.

SOTOMAYOR, A. et al. Forecast urban air pollution in mexico city by using support vector machines: A kernel performance approach. **International Journal of Intelligence Science**, v. 03, n. 03, p. 126–135, 2013. Disponível em: <<http://www.dx.doi.org/10.4236/ijis.2013.33014>>.

TICAY, J. R. et al. 2012. Disponível em: <<http://www.intechopen.com/books/export/citation/BibTex/biodiversity-conservation-and-utilization-in-a-diverse-world/image-processing-for-spider-classification>>.

VAPNIK, V. N. **The nature of statistical learning theory**. New York, NY, USA: Springer-Verlag New York, Inc., 1995. ISBN 0-387-94559-8.

- VETON, K. 2011. Disponível em: <<http://www.intechopen.com/books/export/citation/-BibTex/speech-technologies/wake-up-word-speech-recognition>>.
- VIDHYA, M. Efficient classification of portscan attacks using support vector machine. In: **Green High Performance Computing (ICGHPC), 2013 IEEE International Conference on**. [S.l.: s.n.], 2013. p. 1–5.
- WANG, M.; LI, W.; WANG, X. Transferring a generic pedestrian detector towards specific scenes. In: **Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on**. [S.l.: s.n.], 2012. p. 3274–3281. ISSN 1063-6919.
- WANG, X.; WANG, M.; LI, W. Scene-specific pedestrian detection for static video surveillance. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, v. 36, n. 2, p. 361–374, 2014. ISSN 0162-8828.
- WU, Y.-H.; LIN, S.-J.; YANG, D.-L. A mobile emotion recognition system based on speech signals and facial images. In: **Computer Science and Engineering Conference (ICSEC), 2013 International**. [S.l.: s.n.], 2013. p. 212–217.
- XU, X. et al. Human behavior understanding for video surveillance: Recent advance. In: **Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on**. [S.l.: s.n.], 2010. p. 3867–3873. ISSN 1062-922X.
- YOO, P. et al. Tree-based consensus model for proline cis-trans isomerization prediction. In: **Parallel and Distributed Processing Symposium Workshops PhD Forum (IPDPSW), 2013 IEEE 27th International**. [S.l.: s.n.], 2013. p. 454–458.
- ZHANG, C. et al. Recognizing human group behaviors with multi-group causalities. In: **Web Intelligence and Intelligent Agent Technology (WI-IAT), 2012 IEEE/WIC/ACM International Conferences on**. [S.l.: s.n.], 2012. v. 3, p. 44–48.