



**RECONHECIMENTO DE AÇÕES EM VÍDEO
UTILIZANDO DESCRITORES DE PONTOS
DE INTERESSE ESPAÇO-TEMPORAIS (STIPS)**

ANA PAULA GONÇALVES SOARES DE ALMEIDA

**DISSERTAÇÃO DE MESTRADO EM SISTEMAS MECATRÔNICOS
DEPARTAMENTO DE ENGENHARIA MECÂNICA**

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA MECÂNICA**

**RECONHECIMENTO DE AÇÕES EM VÍDEO
UTILIZANDO DESCRITORES DE PONTOS
DE INTERESSE ESPAÇO-TEMPORAIS (STIPS)**

ANA PAULA GONÇALVES SOARES DE ALMEIDA

Orientador: PROF. DR. FLÁVIO DE BARROS VIDAL, CIC/UNB

DISSERTAÇÃO DE MESTRADO EM SISTEMAS MECATRÔNICOS

**PUBLICAÇÃO PPMEC.DM - 112/2017
BRASÍLIA-DF, 30 DE JANEIRO DE 2017.**

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA MECÂNICA**

**RECONHECIMENTO DE AÇÕES EM VÍDEO
UTILIZANDO DESCRITORES DE PONTOS
DE INTERESSE ESPAÇO-TEMPORAIS (STIPS)**

ANA PAULA GONÇALVES SOARES DE ALMEIDA

DISSERTAÇÃO DE MESTRADO ACADÊMICO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA MECÂNICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM SISTEMAS MECATRÔNICOS.

APROVADA POR:

Prof. Dr. Flávio de Barros Vidal, CIC/UnB
Orientador/Presidente

Prof. Dr. Alexandre Ricardo Soares Romariz, ENE/UnB
Examinador Externo

Prof. Dr. Carlos Humberto Llanos Quintero, ENM/UnB
Examinador Interno

Profa. Dra. Andrea Cristina dos Santos, ENM/UnB
Examinador Interno (Suplente)

BRASÍLIA, 30 DE JANEIRO DE 2017.

FICHA CATALOGRÁFICA

ANA PAULA GONÇALVES SOARES DE ALMEIDA

Reconhecimento de Ações em Vídeo utilizando Descritores de Pontos de Interesse Espaço-Temporais (STIPs)

2017xv, 99p., 201x297 mm

(ENM/FT/UnB, Mestre, Sistemas Mecatrônicos, 2017)

Dissertação de Mestrado - Universidade de Brasília

Faculdade de Tecnologia - Departamento de Engenharia Mecânica

REFERÊNCIA BIBLIOGRÁFICA

ANA PAULA GONÇALVES SOARES DE ALMEIDA (2017) Reconhecimento de Ações em Vídeo utilizando Descritores de Pontos de Interesse Espaço-Temporais (STIPs). Dissertação de Mestrado em Sistemas Mecatrônicos, Publicação 112/2017, Departamento de Engenharia Mecânica, Universidade de Brasília, Brasília, DF, 99p.

CESSÃO DE DIREITOS

AUTOR: Ana Paula Gonçalves Soares de Almeida

TÍTULO: Reconhecimento de Ações em Vídeo utilizando Descritores de Pontos de Interesse Espaço-Temporais (STIPs).

GRAU: Mestre ANO: 2017

É concedida à Universidade de Brasília permissão para reproduzir cópias desta dissertação de Mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor se reserva a outros direitos de publicação e nenhuma parte desta dissertação de Mestrado pode ser reproduzida sem a autorização por escrito do autor.

Ana Paula Gonçalves Soares de Almeida

Agradecimentos

Agradeço primeiramente e principalmente aos meus pais Gorete e Rafael, pois sem eles nenhum trabalho poderia ser feito. Me alegram quando estou desanimada, me apoiam quando estou focada e me embalam quando estou cansada. Eu amo vocês!

Agradeço aos meus familiares, em especial à: minha vó Zezé, minha vó Irene, minhas tias Ester e Graciene, minhas primas Ivy e Letícia, por me mostrarem à força e a capacidade inegáveis que a mulher tem em qualquer carreira que ela desejar seguir, vocês são um exemplo! E aos meus tios Gerson e Carlinhos por as apoiarem em todas as situações.

Ao meu namorado, André Bauer, por me ajudar todas às vezes que precisei e por estar comigo durante toda essa caminhada, segurando minha mão e dizendo que tudo ia dar certo.

Agradeço ao professores que tive ao longo da minha graduação e mestrado, professores Alexandre Zaghetto, Bruno Macchiavello, Li Weigang, Pedro Berger, Ricardo Queiroz, Alexandre Romariz, Leonardo Menezes, Stefan Blawid e Andrea dos Santos por me passarem seus conhecimentos de maneira tão emblemática a ponto de me incentivar a continuar estudando cada vez mais.

Aos meus colegas de pós, Luiz e Lucas, pelas conversas e apoio mútuo. Aos meus amigos de coração: Rebeca e Pedro, obrigada pelas visitas, pelos jogos, pelos presentes, pelas risadas. Marcos, João, Lucas e Paula, que estão comigo desde a escola e são tão brilhantes que me orgulham todos os dias. Ciro, pelo apoio, mensagens de motivação e os melhores links. Kelly, Mabel e Martha, me mostrando que as amizades "mais novas" podem ser tão íntimas quanto amizades de décadas. Victor, Tiago, Murilo e Phillipe, pelas saídas quase semanais e por me fazerem rir tanto e espairecer. Obrigada pela amizade!

E, por fim, mas não menos importante, ao meu orientador e amigo Prof. Flávio Vidal. Obrigada por acreditar em mim e achar que eu sempre posso fazer mais do que o previsto, por me passar seu conhecimento, por me ajudar em todas as burocracias, por me incentivar e me mostrar o caminho, por ser ético e fazer tudo do jeito certo. Eu me espelho em você e tenho muito orgulho em ser sua aluna, obrigada por me acolher.

RESUMO

Nas últimas três décadas, o reconhecimento de ações humanas em vídeo se tornou um tópico amplamente estudado na visão computacional e várias técnicas foram apresentadas para solucionar esse problema com robustez e eficiência. Dentre essas técnicas, os trabalhos que utilizam descritores com características locais espaço-temporais chamam a atenção por terem a capacidade de fazer o reconhecimento em ambientes não-controlados, ou seja, ambientes próximos ao do mundo real. Neste trabalho são avaliadas duas técnicas de pontos de interesse espaço-temporais, uma sendo o estado-da-arte e uma evolução da primeira, para o reconhecimento de ações humanas em sequências de imagens. Estas são colocadas frente a frente, comparando os parâmetros de configuração e classificando a matriz de pontos obtidos de modo que o reconhecimento de ações tanto em bases de vídeos complexas quanto em bases simples possa ser realizado. A metodologia proposta utiliza os pontos de interesse em sua forma pura, como um descritor, uma abordagem inédita de ambas as técnicas apresentadas, bem como realizando a classificação com três tipos de classificadores distintos demonstrando a robustez e eficiência exigidas no processo de reconhecimento de ações em vídeo.

ABSTRACT

Over the last three decades, human action recognition has become a widely studied topic in computer vision and several techniques have been presented to solve this problem in a robust and effective way. Among these techniques, the works that use local spatio-temporal characteristics draw attention because they have the capacity to recognize human action in uncontrolled environments, that is, environments that are similar to the real world. In this work, two techniques of spatio-temporal points of interest are presented, one in state-of-the-art and an evolution of the first, for the recognition of human actions in sequences of images. They are placed face to face, comparing the configuration parameters and classifying the obtained points matrix so that the recognition of actions both in complex bases and in simple bases can be performed. The proposed methodology uses the interest points in its pure form as a descriptor, an unseen approach, not even by the main author of both techniques presented, and classified them with three distinct classifiers, showing the robustness and efficiency required in the process of action recognition in video.

SUMÁRIO

1	INTRODUÇÃO	1
2	RECONHECIMENTO DE AÇÕES EM VÍDEOS	3
2.1	RECONHECIMENTO BASEADO EM MODELO	3
2.2	RECONHECIMENTO BASEADO EM APARÊNCIA	5
3	DESCRITORES BASEADOS EM PONTOS DE INTERESSE ESPAÇO-TEMPORAIS	9
3.1	TRABALHOS RELACIONADOS	9
3.2	PONTOS DE INTERESSE NO ESPAÇO-TEMPO	12
3.2.1	SEQUÊNCIAS DE IMAGENS	12
3.2.2	DETECTOR DE BORDAS DE HARRIS	12
3.2.3	DETECÇÃO DOS PONTOS DE INTERESSE.....	14
3.3	PONTOS DE INTERESSE NO ESPAÇO-TEMPO COM ADAPTAÇÃO DE VELOCIDADE E ESCALA	16
3.3.1	ADAPTAÇÃO DE ESCALA.....	16
3.3.2	ADAPTAÇÃO DE VELOCIDADE	18
4	METODOLOGIA	21
4.1	ENTRADA	21
4.1.1	KTH	22
4.1.2	UCF101.....	23
4.1.3	WEIZMANN	23
4.1.4	YOUTUBE	24
4.2	EXTRAÇÃO DOS DESCRITORES STIP	24
4.2.1	C-STIP.....	24
4.2.2	V-STIP	25
4.2.3	ANÁLISE DA COMPLEXIDADE DOS ALGORITMOS	26
4.3	CLASSIFICAÇÃO DAS AÇÕES	26
4.3.1	MÁQUINAS DE VETORES DE SUPORTE.....	26
4.3.2	REDES NEURAS ARTIFICIAIS.....	28
4.3.3	CONJUNTOS DE DADOS	28
4.4	ANÁLISE DA PRECISÃO	29

5	RESULTADOS.....	31
5.1	RESULTADOS DA CLASSIFICAÇÃO UTILIZANDO SVM.....	32
5.2	RESULTADOS DA CLASSIFICAÇÃO UTILIZANDO RNA.....	38
5.2.1	REDES NEURAS DE RECONHECIMENTO DE PADRÕES.....	38
5.2.2	REDES NEURAS DE AJUSTES DE FUNÇÕES.....	41
5.3	DISCUSSÕES DOS RESULTADOS.....	46
6	CONCLUSÃO.....	49
	REFERÊNCIAS BIBLIOGRÁFICAS.....	51
6.1	MATRIZES DE CONFUSÃO.....	56
6.2	CURVAS ROC.....	72
6.2.1	CURVAS ROC DA REDE NEURAL DE RECONHECIMENTO DE PADRÕES.....	72
6.2.2	CURVAS ROC DA REDE NEURAL DE AJUSTES DE FUNÇÕES.....	80
6.3	CURVAS R.....	88
6.4	REDES NEURAS ARTIFICIAIS.....	96
6.4.1	PERCEPTRONS DE CAMADA ÚNICA.....	96
6.4.2	PERCEPTRONS DE MÚLTIPLAS CAMADAS.....	98
6.5	MÁQUINAS DE VETORES DE SUPORTE.....	99

LISTA DE FIGURAS

2.1	Modelo para interpretação de imagens, baseado nos trabalhos de [1] e [2].	4
2.2	Modelo tridimensional do corpo humano. Retirado de [2].	5
2.3	Modelo de corpo 3D apresentado por [3].	5
2.4	Quadros 0, 13, 20, 30 e 40 de uma pessoa sentando e as respectivas imagens cumulativas de movimento. Retirado de [4].	6
2.5	Combinação entre um quadro chave e um quadro em uma sequência de imagens. Retirado de [5].	7
2.6	STV gerado aplicando o método de correspondência de pontos. Retirado de [6].	8
3.1	Extração de cuboides a partir dos pontos de interesse. Retirado de [7].	10
3.2	Detecção dos pontos de saliência espaço-temporais. Adaptado de [8].	11
3.3	Detecção dos pontos espaço-temporais com informações globais. Retirado de [9].	11
3.4	STIPs Hessianos, com densidade das características variando de muito esparsos (primeira e terceira imagem) a muito densos (segunda e quarta). Retirado de [10].	11
3.5	STIPs Seletivos. Retirado de [11].	12
3.6	Detector de Harris.	14
3.7	Comparação entre a obtenção dos pontos de interesse espaciais e espaço-temporais em uma sequência de vídeo, retirado de [12].	16
3.8	STIP com adaptação de escala, retirado de [13].	18
4.1	Fluxograma da metodologia proposta.	21
5.1	Curvas ROC do método C-STIP para o cenário de treinamento CT_6 utilizando a classificação de redes neurais de reconhecimento de padrões.	39
5.2	Curvas ROC do método V-STIP para o cenário de treinamento CT_1 utilizando a classificação de redes neurais de reconhecimento de padrões.	40
5.3	Curvas ROC do método C-STIP para o cenário de treinamento CT_6 utilizando a classificação de redes neurais de ajustes de funções.	41
5.4	Curvas ROC do método V-STIP para o cenário de treinamento CT_1 utilizando a classificação de redes neurais de ajustes de funções.	42

5.5	Comparação da classificação de C-STIP entre Rede Neural para Reconhecimento de Padrões e para Ajustes de Funções.	43
5.6	Comparação da classificação de V-STIP entre Rede Neural para Reconhecimento de Padrões e para Ajustes de Funções.	44
5.7	Comparação da classificação de C-STIP entre Rede Neural para Ajustes de Funções e SVM.	45
5.8	Comparação da classificação de V-STIP entre Rede Neural para Ajustes de Funções e SVM.	46
6.1	Matriz de confusão do CT_1 para a base de dados KTH.	56
6.2	Matriz de confusão do CT_2 para a base de dados KTH.	57
6.3	Matriz de confusão do CT_3 para a base de dados KTH.	57
6.4	Matriz de confusão do CT_4 para a base de dados KTH.	58
6.5	Matriz de confusão do CT_5 para a base de dados KTH.	58
6.6	Matriz de confusão do CT_6 para a base de dados KTH.	59
6.7	Matriz de confusão do CT_7 para a base de dados KTH.	59
6.8	Matriz de confusão do CT_1 para a base de dados UCF101.....	60
6.9	Matriz de confusão do CT_2 para a base de dados UCF101.....	60
6.10	Matriz de confusão do CT_3 para a base de dados UCF101.....	61
6.11	Matriz de confusão do CT_4 para a base de dados UCF101.....	61
6.12	Matriz de confusão do CT_5 para a base de dados UCF101.....	62
6.13	Matriz de confusão do CT_6 para a base de dados UCF101.....	62
6.14	Matriz de confusão do CT_7 para a base de dados UCF101.....	63
6.15	Matriz de confusão do CT_1 para a base de dados Weizmann.....	64
6.16	Matriz de confusão do CT_2 para a base de dados Weizmann.....	64
6.17	Matriz de confusão do CT_3 para a base de dados Weizmann.....	65
6.18	Matriz de confusão do CT_4 para a base de dados Weizmann.....	65
6.19	Matriz de confusão do CT_5 para a base de dados Weizmann.....	66
6.20	Matriz de confusão do CT_6 para a base de dados Weizmann.....	66
6.21	Matriz de confusão do CT_7 para a base de dados Weizmann.....	67
6.22	Matriz de confusão do CT_1 para a base de dados YouTube.....	68
6.23	Matriz de confusão do CT_2 para a base de dados YouTube.....	68
6.24	Matriz de confusão do CT_3 para a base de dados YouTube.....	69
6.25	Matriz de confusão do CT_4 para a base de dados YouTube.....	69
6.26	Matriz de confusão do CT_5 para a base de dados YouTube.....	70
6.27	Matriz de confusão do CT_6 para a base de dados YouTube.....	70
6.28	Matriz de confusão do CT_7 para a base de dados YouTube.....	71
6.29	Curvas ROC do método C-STIP da base de dados KTH utilizando a classificação de redes neurais de reconhecimento de padrões.	72
6.30	Curvas ROC do método V-STIP da base de dados KTH utilizando a classificação de redes neurais de reconhecimento de padrões.	73

6.31	Curvas ROC do método C-STIP da base de dados UCF101 utilizando a classificação de redes neurais de reconhecimento de padrões.	74
6.32	Curvas ROC do método V-STIP da base de dados UCF101 utilizando a classificação de redes neurais de reconhecimento de padrões.	75
6.33	Curvas ROC do método C-STIP da base de dados Weizmann utilizando a classificação de redes neurais de reconhecimento de padrões.	76
6.34	Curvas ROC do método V-STIP da base de dados Weizmann utilizando a classificação de redes neurais de reconhecimento de padrões.	77
6.35	Curvas ROC do método C-STIP da base de dados YouTube utilizando a classificação de redes neurais de reconhecimento de padrões.	78
6.36	Curvas ROC do método V-STIP da base de dados YouTube utilizando a classificação de redes neurais de reconhecimento de padrões.	79
6.37	Curvas ROC do método C-STIP da base de dados KTH utilizando a classificação de redes neurais de ajustes de funções.	80
6.38	Curvas ROC do método V-STIP da base de dados KTH utilizando a classificação de redes neurais de ajustes de funções.	81
6.39	Curvas ROC do método C-STIP da base de dados UCF101 utilizando a classificação de redes neurais de ajustes de funções.	82
6.40	Curvas ROC do método V-STIP da base de dados UCF101 utilizando a classificação de redes neurais de ajustes de funções.	83
6.41	Curvas ROC do método C-STIP da base de dados Weizmann utilizando a classificação de redes neurais de ajustes de funções.	84
6.42	Curvas ROC do método V-STIP da base de dados Weizmann utilizando a classificação de redes neurais de ajustes de funções.	85
6.43	Curvas ROC do método C-STIP da base de dados YouTube utilizando a classificação de redes neurais de ajustes de funções.	86
6.44	Curvas ROC do método V-STIP da base de dados YouTube utilizando a classificação de redes neurais de ajustes de funções.	87
6.45	Curvas R do método C-STIP da base de dados KTH utilizando a classificação de redes neurais de ajustes de funções.....	88
6.46	Curvas R do método V-STIP da base de dados KTH utilizando a classificação de redes neurais de ajustes de funções.....	89
6.47	Curvas R do método C-STIP da base de dados UCF101 utilizando a classificação de redes neurais de ajustes de funções.	90
6.48	Curvas R do método V-STIP da base de dados UCF101 utilizando a classificação de redes neurais de ajustes de funções.	91
6.49	Curvas R do método C-STIP da base de dados Weizmann utilizando a classificação de redes neurais de ajustes de funções.	92
6.50	Curvas R do método V-STIP da base de dados Weizmann utilizando a classificação de redes neurais de ajustes de funções.	93

6.51	Curvas R do método C-STIP da base de dados YouTube utilizando a classificação de redes neurais de ajustes de funções.....	94
6.52	Curvas R do método V-STIP da base de dados YouTube utilizando a classificação de redes neurais de ajustes de funções.....	95
6.53	Modelo de neurônio não-linear. Retirado de [14].....	97
6.54	Hiperplano como fronteira de decisão para a classificação de duas classes. Retirado de [14].....	98
6.55	Hiperplano ótimo. Retirado de [14].....	99

LISTA DE TABELAS

4.1	Amostras das bases de dados.	22
5.1	Configurações utilizadas para processamento dos dados.	31
5.2	Cenários de Treinamento.	32
5.3	Diagonais destacadas da base KTH.	33
5.4	Diagonais destacadas da base UCF101.....	34
5.5	Diagonais destacadas da base Weizmann.	35
5.6	Diagonais destacadas da base YouTube.....	36
5.7	CT_1 - Mean Average Precision (MAP)	36
5.8	CT_2 - Mean Average Precision (MAP)	36
5.9	CT_3 - Mean Average Precision (MAP)	37
5.10	CT_4 - Mean Average Precision (MAP)	37
5.11	CT_5 - Mean Average Precision (MAP)	37
5.12	CT_6 - Mean Average Precision (MAP)	37
5.13	CT_7 - Mean Average Precision (MAP)	37
5.14	Comparação entre as acurácias da base de dados KTH para os trabalhos de Laptev [12] (estado-da-arte), Oikonomopoulos [8], Dollar[7], Willems [10] e Wong [9].....	48

LISTA DE TERMOS E SIGLAS

C-STIP	STIP Clássico
DTW	Dynamic Time Warping
fn	Falso Negativo
fp	Falso Positivo
FPS	Quadros por Segundo
MAP	Precisão Média
MEI	Imagens de Movimento de Energia
MHI	Imagens de Histórico de Movimento
MLP	Perceptrons de Múltiplas Camadas
RNA	Redes Neurais Artificiais
ROC	Característica de Operação do Receptor
SIP	Pontos de Interesse no Espaço
STIP	Pontos de Interesse no Espaço-Tempo
STV	Volume Espaço-Tempo
SVM	Máquinas de Vetores de Suporte
tn	Verdadeiro Negativo
tp	Verdadeiro Positivo
V-STIP	STIP com Adaptação de Velocidade e Escala

Capítulo 1

Introdução

O reconhecimento de ações humanas em vídeo é uma área amplamente estudada nas últimas três décadas, tendo diversos métodos e técnicas distintas [15, 16, 2, 17] aprimorados de forma gradativa neste campo de pesquisa.

Segundo Gorelick [18], o reconhecimento de ações humanas pode ser definido como um componente chave para uma variedade de aplicações em visão computacional, como: videomonitoramento [19, 20], interface humano-computador [21, 22], indexação e navegação de vídeos [19, 21, 22], reconhecimento de gestos [9], análise de eventos esportivos e coreografia de danças [18].

Inúmeras abordagens foram desenvolvidas nos últimos anos, entretanto a maioria possui limitações computacionais [18], sendo estas: dificuldade em estimar o padrão de movimento [23], problemas na abertura da câmera, descontinuidades e superfícies suavizadas [24]; todos relacionados à maneira utilizada para estimar o movimento, desde fluxo ótico a técnicas mais complexas, como autoformas de silhuetas de primeiro plano, descritas no trabalho de Goldenberg [25].

Tais problemas ocorrem também devido à grande parte dos trabalhos supracitados basearem-se na computação de gradientes locais espaço-temporais ou em características de intensidade, possuindo resultados dúbios em casos de vídeos com baixa qualidade, descontinuações de movimento ou serrilhamentos temporais [18].

Em trabalhos recentes bem sucedidos, descritos por Chakraborty [11] e Dehghan [26], informações de sequências de vídeo como volumes de intensidades [6], gradientes [27], fluxo ótico [28] ou outros tipos de características locais no espaço-tempo são utilizadas para realizar o reconhecimento de ações humanas em vídeos. Entretanto, a presença de outros movimentos perto da região onde ocorre a ação atrapalha a classificação de Chakraborty [11] e além de usar poucas bases para avaliar o método, Dehghan [26] usa anotações manuais para validar suas informações, tornando imprecisos os resultados obtidos.

A partir das deficiências dos trabalhos anteriores, o principal trabalho que explora as características locais espaço-temporais é apresentado por Laptev [17], em que os pontos de

interesse espaço-temporais são conceituados e baseia outros textos, como os de Dollar [7], Laptev [29], Willems [10] e Chakraborty [11], que complementam o método com técnicas, como Histogramas de Gradientes Orientados (HoOG) [27].

A partir das motivações apresentadas e pela importância do tema, o objetivo deste trabalho é avaliar os pontos de interesse espaço-temporais (C-STIP) propostos por Laptev [17], e sua evolução, apresentada no trabalho de Laptev [29], que trata-se dos pontos de interesse no espaço-tempo com adaptação de velocidade e escalas (V-STIP) e classificar esses pontos sem o uso de descritores alternativos (ou adicionais), comprovando a possibilidade de utilizá-los como descritores locais para a tarefa de reconhecimento de ações humanas em vídeos e/ou sequências de imagens digitais.

Uma proposta similar, e utilizada como inspiração apresentada neste trabalho, é mostrada por Schuldt [30], onde há a introdução da versão clássica dos pontos de interesse no espaço-tempo e uma classificação por máquinas de vetores de suporte (*Support Vector Machine* – SVM). Entretanto, em Schuldt [30], apenas uma combinação entre os pontos de interesse e histogramas de características locais é usada para realizar o reconhecimento das ações em vídeo. Esta limitação mostra que para bases de dados pequenas a abordagem tem um bom funcionamento, porém, para bases de dados maiores e mais complexas, um descritor espaço-temporal global é necessário.

Também neste trabalho, uma versão alternativa de utilizar o método do estado-da-arte existente para realizar o reconhecimento de ações humanas em vídeos será apresentado, de modo que a metodologia final empregada siga etapas diferentes de todos os trabalhos relacionados. Esta diferença é que estes métodos não utilizam o STIP em seu estado natural, sempre aplicando modificações, permitindo a criação da primeira contribuição deste trabalho: verificar a capacidade de discriminação de descritores espaço-temporais sem o uso de descritores auxiliares. Em linhas gerais, será a transformação da abordagem do uso destes descritores com características locais para globais no processo de reconhecimento de ações em vídeos.

Ademais, traz-se neste trabalho também como contribuição, a comparação dos dois métodos diante um do outro, avaliando sua capacidade de reconhecimento de ações em vídeos. Os subconjuntos das bases usadas em testes são classificadas com três versões distintas de classificadores: Máquinas de Vetores de Suporte, Perceptrons de Múltiplas Camadas para classificação de padrões e Perceptrons de Múltiplas Camadas para ajustes de funções. Os classificadores usados foram escolhidos de modo que uma classificação simples, como o SVM, pudesse ser comparada com uma classificação mais robusta, como as redes neurais.

O Capítulo 2 apresenta trabalhos relacionados à bibliografia básica requerida para que a metodologia seja corretamente interpretada. O Capítulo 3 contém uma explicação detalhada das técnicas de pontos de interesse no espaço-tempo usadas. No Capítulo 4, a metodologia proposta é mostrada. Os Capítulos 5 e 6 são dedicados à exposição e discussão dos resultados e conclusão e trabalhos futuros, respectivamente.

Capítulo 2

Reconhecimento de Ações em Vídeos

Neste capítulo serão apresentados os principais conceitos e trabalhos relativos ao reconhecimento de ações humanas em vídeos, baseados em uma linha histórica de aparecimento na bibliografia utilizada como referência.

O reconhecimento de ações humanas é o processo de classificação de eventos ocorrendo em um vídeo ou em sequências de imagens [31]. As aplicações para o reconhecimento de ações humanas são variadas. O trabalho de Niebles [19] utiliza o reconhecimento de ações para realizar o videomonitoramento, localizando e categorizando automaticamente indivíduos em câmeras.

Em Zhang [20], a câmera de um robô é usada para capturar a profundidade de vídeos e utilizar estas informações para fazer o reconhecimento de ações, criando uma interface humano-máquina. De acordo com Blank [32], a análise generalizada das formas bidimensionais permite efetuar o reconhecimento de ações para encontrar movimentos de dança em sequências de vídeo e explorar cenas de esporte.

Usualmente, o reconhecimento de ações é dividido entre duas grandes vertentes, de acordo com Bobick [4] e Laptev [33]: reconhecimento baseado em modelo e reconhecimento baseado em aparência.

2.1 Reconhecimento Baseado em Modelo

Um modelo geral para interpretação de imagens foi proposto por Kanade [1]. Dada uma imagem, características do domínio da figura ou da cena são extraídas, sendo depois utilizadas para acessar o modelo genérico que, por sua vez, gera uma hipótese. Esta hipótese é verificada ao se projetar no nível da figura, combinando-a com a imagem de entrada. Este ciclo está representado na Figura 2.2.

Uma imagem é uma projeção da cena [1], enquanto uma cena é uma organização de corpos (e suas superfícies, bordas e cantos) em relação a um ponto de vista [34]. Neste mo-

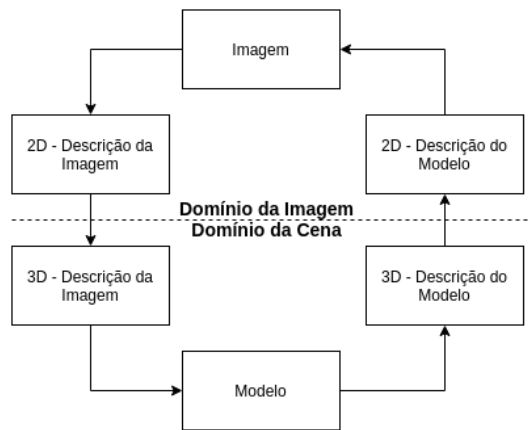


Figura 2.1: Modelo para interpretação de imagens, baseado nos trabalhos de [1] e [2].

delo, o domínio da imagem contém todas as características que se referem apenas à imagem, como segmentos de linha, regiões homogêneas e intensidades de gradiente. O domínio da cena possui características exclusivas da cena, como orientações de superfície, refletância e configurações de bordas [35]. Sendo assim, esta abordagem requer uma segmentação robusta tanto de fundo (*background*) quanto de primeiro plano (*foreground*) para que seja possível fazer a distinção entre os domínios apresentados.

O modelo supracitado é utilizado por Rohr [2], que afirma que um protótipo humano é comumente obtido com a recriação do corpo humano. Para tal, usa-se uma representação tridimensional com graus de liberdade que permitem a formação de poses distintas, representando um certo movimento que corresponde a uma ação.

Os graus de liberdade possuem uma forte relação com o número de movimentos representados. Quanto mais graus de liberdade, uma maior quantidade de posições de corpo pode ser adquirida, criando uma variedade de movimentos diferentes e, conseqüentemente, de ações representadas. Para realizar o reconhecimento, um modelo humano 3D é desenhado a partir de formas cilíndricas (Figura 2.2) e um Filtro de Kalman [36] é usado para estimar os parâmetros do modelo.

Segundo Gravila [15], a habilidade de reconhecer humanos e suas atividades por imagens é essencial para a criação de uma máquina que tenha a capacidade fazer a interação computador-humano de forma inteligente. Para fazer o reconhecimento, a abordagem de Gravila [15] usa dois elementos principais: recuperação e rastreamento da posição corporal e reconhecimento dos padrões de movimento.

A Figura 2.3 mostra um modelo de corpo tridimensional reconstruído a partir de sequências de imagens adquiridas de múltiplas câmeras estacionárias, previamente calibradas. O modelo, que possui 17 graus de liberdade, é utilizado como a entrada de um componente de reconhecimento de movimento. A técnica usada para o padrão de movimentos foi o *Dynamic Time Warping* (DTW).

Em Rehg [37], um modelo cinemático é usado para prever oclusões, além de padrões com funções de janelamento que auxiliam no rastreamento de objetos parcialmente ocultos.

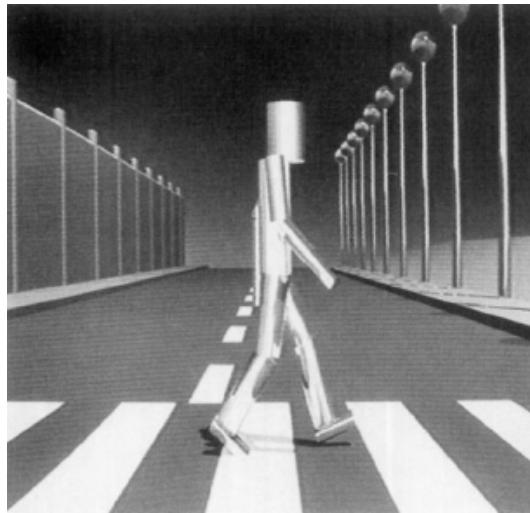


Figura 2.2: Modelo tridimensional do corpo humano. Retirado de [2].

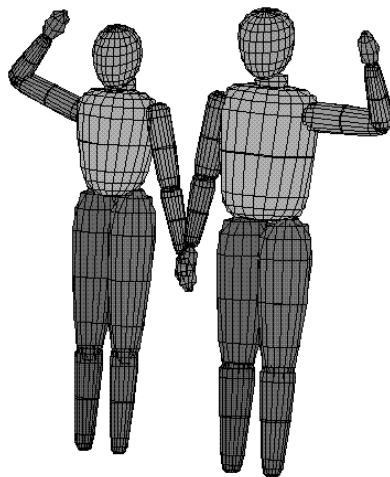


Figura 2.3: Modelo de corpo 3D apresentado por [3].

Outrossim, em Canton-Ferrer [38], é utilizado um modelo elipsoide de corpo humano que se adapta a uma entrada 3D, de modo que essa adaptação mostre em qual parte do corpo o gesto ocorre, aumentando assim a capacidade de reconhecimento do sistema como um todo e gerando uma saída mais robusta para o classificador Bayesiano.

De acordo com Bobick [4], uma vantagem do reconhecimento baseado em modelo é uma maneira mais simplificada de estimar e prever a localização de características.

2.2 Reconhecimento Baseado em Aparência

O reconhecimento baseado em aparência, ou temporal, enfatiza a representação da ação como um movimento ao longo do tempo e o reconhecimento deste movimento é obtido a partir da aparência apresentada, visto que há o delineamento de uma trajetória no espaço-tempo [4].

Quando há a existência de borrões nas sequências de imagens, impossibilitando a visualização da ocorrência de um evento, não é possível utilizar estruturas tridimensionais do corpo humano para fazer o reconhecimento da ação. Portanto, a necessidade de reconhecer apenas a partir do movimento se faz presente.

No trabalho de Davis [16], um vetor de imagem, ou um padrão temporal, é construído de forma que ele possa ser comparado com representações de ações já conhecidas.



(a) Vídeo de referência.



(b) Padrões temporais do vídeo de referência.

Figura 2.4: Quadros 0, 13, 20, 30 e 40 de uma pessoa sentando e as respectivas imagens cumulativas de movimento. Retirado de [4].

A Figura 2.4(a) apresenta uma sequência de imagens que corresponde à uma pessoa realizando um movimento de sentar-se. A Figura 2.4(b) mostra as imagens de movimento binárias acumuladas (*Motion-energy Images – MEI*), chamadas de imagens de energia de movimento. Nota-se que a região branca da Figura 2.4(b) corresponde à área em que ocorre movimento. O formato dessa região sugere tanto a ação que ocorreu quanto o ângulo de visão. Para representar como a ação da imagem está se movimentando, imagens de histórico de movimento são formadas (*Motion-history Images – MHI*).

Dado um conjunto de MEIs e MHIs para cada combinação de movimento, descritores estatísticos dessas imagens são computados usando características baseadas em momento. Um modelo estatístico dos momentos (média e matriz de covariância) é gerado para tanto para o MEI quanto para o MHI. Para reconhecer uma ação de entrada, a distância de Mahalanobis é calculada entre a descrição do momento da entrada e cada um dos movimentos conhecidos.

Entretanto, o plano de fundo da cena deve ser estático e quando a ação apresenta pouco movimento em certas partes do corpo, como o lançamento de uma bola com as pernas paradas, por exemplo, a ação das pernas é determinado pelo próprio movimento, induzindo uma

maior variação na descrição estatística do padrão temporal.

O reconhecimento de ação baseado em aparência também pode ser obtido a partir da análise do movimento da sequência de imagens. Em [28], o campo de movimento do fluxo ótico é usado para realizar a segmentação da cena e auxiliar na detecção e rastreamento de objetos em movimento. Entretanto, o uso desta técnica pode afetar a performance da abordagem, uma vez que o fluxo ótico não se mostra eficaz com mudanças bruscas de iluminação e movimentação [39].

Segundo Carlsson [5], ações humanas são, em geral, caracterizadas por uma sequência de posturas corporais específicas e a grande maioria destas ações pode ser reconhecida a partir de uma vista única.

Ainda de acordo com Carlsson [5], a forma é representada como bordas, adquiridas a partir do detector de bordas de Canny [40], e o algoritmo de comparação é fundamentado a partir da estimação da deformação da forma da imagem com relação à forma de um quadro chave. A Figura 2.5 apresenta essa comparação.

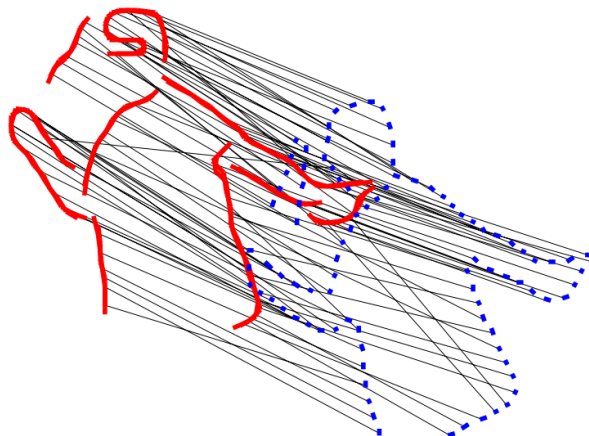
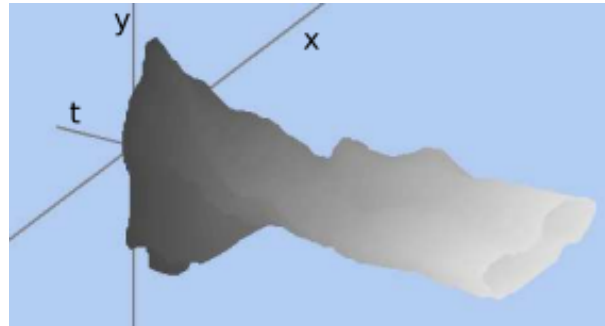


Figura 2.5: Combinação entre um quadro chave e um quadro em uma sequência de imagens. Retirado de [5].

Outra técnica baseada na forma da ação é mostrada em Yilmaz [6]. Uma ação 3D projeta pontos de contorno em 2D no plano da imagem. Uma sequência desses pontos de contorno gera um volume espaço-temporal (*Space-time Volume* – STV), considerando o tempo, e com as características obtidas a partir da análise do STV, descritores de ação são computados. Com esses descritores, o reconhecimento de ação é executado.



(a) Sequência de contornos de objetos rastreados para a ação "cair".



(b) STV da ação "cair".

Figura 2.6: STV gerado aplicando o método de correspondência de pontos. Retirado de [6].

A metodologia descrita em Laptev [17] usa técnicas combinadas baseadas em aparência (*e.g.* [41] e [42]) para a criação de um novo detector de eventos de movimento usando informação local. Esta representação não necessita de segmentação ou rastreamento prévio [43] e é baseada no detector de bordas de Harris [44]. Ademais, o detalhamento desse método está descrito no Capítulo 3.

Ao longo dos anos, múltiplos detectores fundamentados em pontos de interesse espaço-temporais (*Space-Time Interest Points* – STIP) foram propostos a partir da introdução da primeira técnica de STIP de Laptev [17]. Estes serão igualmente explanados no Capítulo 3.

Capítulo 3

Descritores baseados em Pontos de Interesse Espaço-Temporais

Em visão computacional, a detecção de pontos de interesse em uma imagem é amplamente utilizada para resolver problemas de reconhecimento de objetos, rastreamento, reconstrução 3D e reconhecimento de ações. Esta popularidade se dá devido à condensação da área de análise da imagem [45].

3.1 Trabalhos Relacionados

A abordagem de Dollar [7] detecta e caracteriza comportamentos a partir de sequências de vídeo, utilizando pontos de interesse espaço-temporais, onde uma ação é considerada um comportamento.

Entretanto, os STIPs usados não são os mesmos propostos por Laptev [17], pois Dollar [7] considera que os STIPs não são adequados para bases de vídeos com pouca movimentação, uma vez que estas não dão origem a muitas bordas espaço-temporais por terem movimentos sutis e graduais. A abordagem será melhor explicada nos parágrafos seguintes.

Os pontos de interesse utilizam uma função de resposta R_d (Equação 3.1) calculada a partir da adição de filtros lineares separáveis convoluídos a uma máscara do filtro Gaussiano 2D G_d , aplicada apenas nas dimensões espaciais, e a um par de quadraturas de um filtro de Gabor unidimensional H_{ev} e H_{od}

$$R_d = (\mathbf{I} * G_d * H_{ev})^2 + (\mathbf{I} * G_d * H_{od})^2. \quad (3.1)$$

Em cada ponto de interesse, um cuboide contendo valores de *pixels* janelados no espaço-tempo são extraídos e para criar um descritor que pudesse fazer a comparação desses cuboides, a distância Euclidiana foi usada. A Figura 3.1 mostra os cuboides extraídos de uma sequência de imagens.

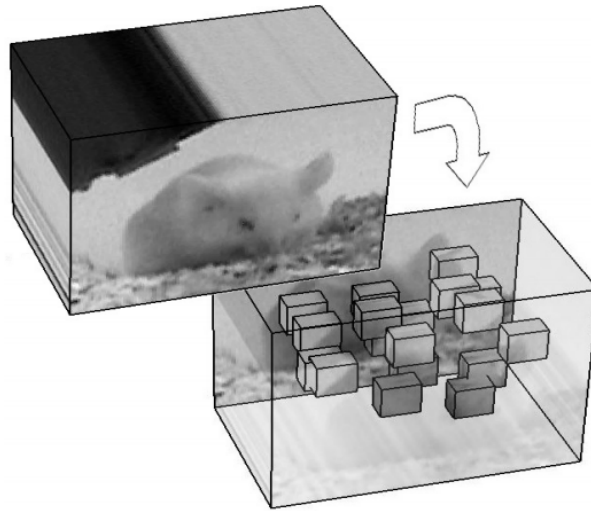


Figura 3.1: Extração de cuboides a partir dos pontos de interesse. Retirado de [7].

Em Laptev [29], uma evolução do primeiro STIP apresentado é descrita, adaptando a velocidade e as características espaço-temporais de escala, de modo que uma representação estável de vídeo seja obtida. A Seção 3.3 mostra esta técnica detalhadamente.

O trabalho de Oikonomopoulos [8] introduz o uso de uma representação esparsa de sequências de imagens como uma coleção de eventos espaço-temporais que são salientes tanto no espaço quanto no tempo. Os pontos de saliência no espaço-tempo são detectados a partir da medição das variações no conteúdo de informações de *pixels* vizinhos no espaço-tempo.

Ao contrário do trabalho de Laptev [17], a representação contém pontos de interesse espaço-temporais onde há picos de variação de atividade, como bordas de um objeto em movimento. As escalas são detectadas automaticamente de acordo com o alcance máximo local de entropia e as regiões de saliência espaço-temporais são agrupamentos de pontos no espaço-tempo com localização e escala similares. Cada sequência de imagem é então representada como um conjunto de pontos de saliência no espaço-tempo.

Para calcular a distância entre duas representações, a distância de Chamfer é utilizada e para lidar com a problemática diferença de velocidade na execução de uma ação, é proposta uma técnica de deformação espaço-temporal linear, que procura diminuir as distâncias de Chamfer. A Figura 3.2 mostra a detecção destes pontos de saliência usando dois objetos distintos realizando a mesma ação.



Figura 3.2: Detecção dos pontos de saliência espaço-temporais. Adaptado de [8].

Para Wong [9], as características locais espaço-temporais ou pontos de interesse proporcionam representações compactas e descritivas para a análise de reconhecimento de movimento. Os pontos são extraídos com base em informações globais de cada vídeo de entrada, como a localização de áreas em movimento, e cuboides são detectados em regiões que possuem uma maior probabilidade de conter movimentação relevante. A Figura 3.3 apresenta a detecção dos pontos em uma sequência de imagens que representa a ação "caminhar".

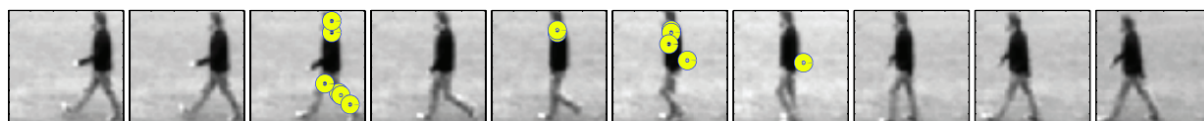


Figura 3.3: Detecção dos pontos espaço-temporais com informações globais. Retirado de [9].

Em Willems [10], os pontos de interesse no espaço-tempo são obtidos a partir do módulo da determinante de uma matriz Hessiana 3D, permitindo que a seleção de escala possa ser realizada de maneiras distintas. O princípio matemático é o mesmo encontrado em Laptev [17] (Demonstrado na Seção 3.2).

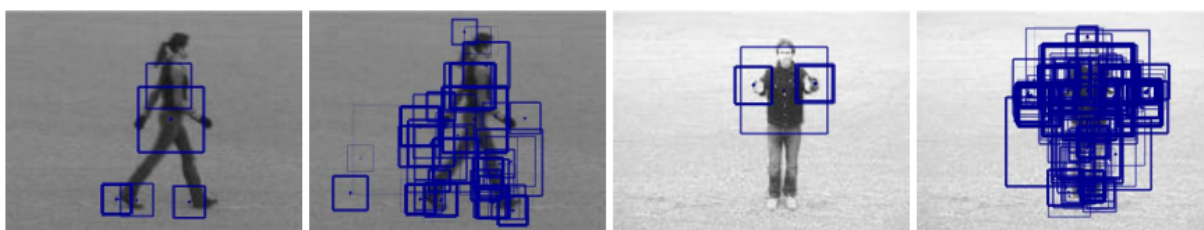


Figura 3.4: STIPs Hessianos, com densidade das características variando de muito esparsa (primeira e terceira imagem) a muito densa (segunda e quarta). Retirado de [10].

Em Chakraborty [11], um método seletivo dos STIPs com supressão de bordaduras associado com contenções locais é apresentado.

Inicialmente, é feita a detecção dos pontos de interesse espaciais (*Space Interest Points* – SIP), com um detector de bordas de Harris. Em seguida, uma anulação dos SIPs do plano de

fundo é feita, utilizando uma máscara de supressão *surround* para cada ponto de interesse. Posteriormente, são impostas restrições locais e temporais.

Esta abordagem inclui um modelo saco-de-vídeos (*Bag of Videos*) para construir um vocabulário de palavras-visuais para o processo de reconhecimento de ações.



Figura 3.5: STIPs Seletivos. Retirado de [11].

3.2 Pontos de Interesse no Espaço-Tempo

De acordo com Laptev [33], o propósito principal dos pontos de interesse espaço-temporais (STIP) é realizar a detecção de eventos diretamente dos dados espaço temporais da imagem, considerando regiões que possuam localidades distintas no espaço-tempo com robustez suficiente para detectar e classificar.

O STIP clássico proposto por Laptev [17], utiliza uma extensão 3D do detector de bordas de Harris [44] para detectar pontos de interesse no espaço-tempo. A seguir serão apresentados os elementos básicos para a construção desse STIP.

3.2.1 Sequências de Imagens

Uma imagem é definida como uma função bidimensional $I_{2d}(x, y)$, onde x e y são coordenadas espaciais e a amplitude de $I_{2d}(x, y)$ em qualquer par de coordenadas (x, y) é chamada de intensidade da imagem neste ponto [46].

De acordo com Trucco [47], uma sequência de imagens é definida por uma série contendo n imagens, chamadas de quadros, obtidas em instantes discretos temporais t . Esta definição será usada como fundamento básico para as demais equações deste trabalho.

3.2.2 Detector de Bordas de Harris

Em Harris [44], a mudança média direcional de intensidade em uma janela em torno de um ponto de interesse é vista como uma borda. Considerando a função janela como $W(x, y)$ e um vetor de deslocamento (u, v) , a variação de intensidade em uma imagem I_{2d} , é dada

pela Equação 3.2, com os limites de (x, y) configurados como as dimensões da imagem.

$$\mathbf{E}(u, v) = \sum_{x,y} \mathbf{W}(x, y) [\mathbf{I}_{2d}(x + u, y + v) - \mathbf{I}_{2d}(x, y)]^2. \quad (3.2)$$

Para encontrar janelas que possuam bordas, uma grande variação na intensidade deve ser encontrada, sendo necessária uma maximização do termo

$$\sum_{x,y} [\mathbf{I}_{2d}(x + u, y + v) - \mathbf{I}_{2d}(x, y)]^2 \quad (3.3)$$

da Equação 3.2 através de uma expansão de Taylor, como apresentado na Equação 3.4.

$$\mathbf{E}(u, v) \approx \sum_{x,y} [\mathbf{I}_{2d}(x, y) + uI_x + vI_y - \mathbf{I}_{2d}(x, y)]^2. \quad (3.4)$$

Expandindo, com I_x como a dimensão horizontal da imagem e I_y como a dimensão vertical da imagem,

$$\mathbf{E}(u, v) \approx \sum_{x,y} u^2 I_x^2 + 2uv I_x I_y + v^2 I_y^2 \quad (3.5)$$

que pode ser expresso em forma matricial como:

$$\mathbf{E}(u, v) \approx \begin{bmatrix} u & v \end{bmatrix} \left(\sum_{x,y} \mathbf{W}(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \right) \begin{bmatrix} u \\ v \end{bmatrix}. \quad (3.6)$$

Para pequenos deslocamentos $[u, v]$, tem-se a aproximação bilinear apresentada na Equação 3.8, onde \mathbf{M} (Equação 3.7) é uma matriz 2×2 computada das imagens derivativas.

$$\mathbf{M} = \sum_{x,y} \mathbf{W}(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}. \quad (3.7)$$

$$\mathbf{E}(u, v) \cong [u, v] \mathbf{M} \begin{bmatrix} u \\ v \end{bmatrix}. \quad (3.8)$$

Então, uma pontuação r (Equação 3.9) é calculada para cada janela, com os autovalores λ_1 e λ_2 e a constante do fator de sensibilidade de Harris k .

$$r = \det(\mathbf{M}) - k(\text{trace}(\mathbf{M}))^2 = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2. \quad (3.9)$$

As Figuras 3.6 e 3.7(a) mostram o funcionamento do detector de Harris em uma imagem e em uma sequência de vídeo, respectivamente.

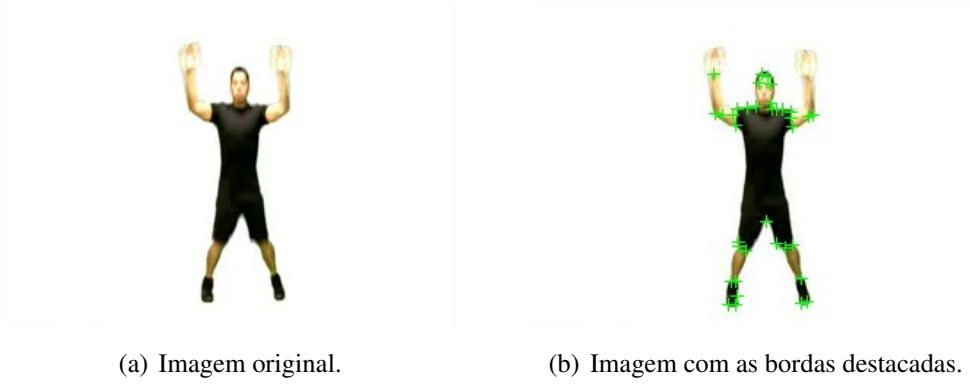


Figura 3.6: Detector de Harris.

3.2.3 Detecção dos Pontos de Interesse

3.2.3.1 Pontos de Interesse Espaciais

Considerando uma escala de observação σ_l^2 e parâmetro de escala σ , os pontos de interesse espaciais na dimensão da espacial imagem, representado pelas dimensões horizontais e verticais, são encontrados a partir de uma matriz de covariância das derivadas Gaussianas \mathbf{L}_x e \mathbf{L}_y convoluída a uma máscara Gaussiana espacial (Equação 3.10) em uma escala $\sigma_i^2 = s\sigma_l^2$.

$$\mathbf{G}(x, y; \sigma^2) = \frac{1}{2\pi\sigma^2} \exp(-(x^2 + y^2)/2\sigma^2). \quad (3.10)$$

Reescrevendo as Equações 3.7 e 3.9 com esses novos termos, temos nas Equações 3.11 e 3.12 a aplicação do detector de bordas de Harris (Subseção 3.2.2) na detecção de pontos de interesse no domínio espacial.

$$\mu = \mathbf{G}(x, y; \sigma_i^2) * \begin{pmatrix} \mathbf{L}_x^2 & \mathbf{L}_x \mathbf{L}_y \\ \mathbf{L}_x \mathbf{L}_y & \mathbf{L}_y^2 \end{pmatrix} \quad (3.11)$$

onde \mathbf{L}_x e \mathbf{L}_y são definidas como

$$\begin{aligned} \mathbf{L}_x(x, y; \sigma_l^2) &= \partial_x(\mathbf{G}((x, y; \sigma_l^2) * \mathbf{I}_{2d}(x, y)), \\ \mathbf{L}_y(x, y; \sigma_l^2) &= \partial_y(\mathbf{G}((x, y; \sigma_l^2) * \mathbf{I}_{2d}(x, y)). \end{aligned} \quad (3.12)$$

$$r = \det(\mu) - k(\text{trace}(\mu))^2 = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2. \quad (3.13)$$

A razão $\alpha = \lambda_2/\lambda_1$ deve ser alta onde há um ponto de interesse. Através da Equação 3.13, a razão α deve satisfazer a condição $k \leq \alpha/(1 + \alpha)^2$ para que seja considerado um local máximo positivo. O fator de sensibilidade k é comumente utilizado como uma constante

$k = 0.04$, porém, valores menores de k permitem uma detecção de pontos de interesse com uma forma mais alongada [12].

3.2.3.2 Pontos de Interesse no Espaço-Tempo

Usando como base a Equação 3.13, pode-se estender a análise espacial para uma espaço-temporal. Para tal, considera-se que há grandes variações de intensidade tanto na dimensão espacial quanto na dimensão temporal [17].

Estes pontos serão pontos de interesse espaciais com uma localização distinta no tempo, correspondendo aos vizinhos espaço-temporais locais com movimento não-constante [12].

Para encontrar os STIPs, a representação linear no espaço-tempo \mathbf{L} de uma sequência de imagem \mathbf{I} é construída a partir da convolução entre uma máscara Gaussiana \mathbf{G}_{st} com a variação do tempo τ_l^2 e variação espacial σ_l^2 .

$$\mathbf{L}(x, y, t; \sigma_l^2, \tau_l^2) = \mathbf{G}_{\text{st}}(x, y, t; \sigma_l^2, \tau_l^2) * \mathbf{I}(x, y, t) \quad (3.14)$$

onde

$$\mathbf{G}_{\text{st}}(x, y, t; \sigma_l^2, \tau_l^2) = \frac{\exp(-(x^2 + y^2)/2\sigma_l^2 - t^2/2\tau_l^2)}{\sqrt{(2\pi)^3 \sigma_l^4 \tau_l^2}}. \quad (3.15)$$

Obtendo a matriz 3×3 de covariâncias das derivadas, com uma função Gaussiana de peso $\mathbf{G}_{\text{st}}(x, y, t; \sigma_l^2, \tau_l^2)$, escalas de integração $\sigma_i^2 = s\sigma_l^2$ e $\tau_i^2 = s\tau_l^2$ e as derivadas de primeira-ordem definidas como $\mathbf{L}_\xi(x, y, t; \sigma_l^2, \tau_l^2) = \partial_\xi(\mathbf{G}_{\text{st}}((x, y, t; \sigma_l^2, \tau_l^2) * \mathbf{I}(x, y, t)$.

$$\mu_{\text{st}} = \mathbf{G}(x, y, t; \sigma_l^2, \tau_l^2) * \begin{pmatrix} \mathbf{L}_x^2 & \mathbf{L}_x \mathbf{L}_y & \mathbf{L}_x \mathbf{L}_t \\ \mathbf{L}_x \mathbf{L}_y & \mathbf{L}_y^2 & \mathbf{L}_y \mathbf{L}_t \\ \mathbf{L}_x \mathbf{L}_t & \mathbf{L}_y \mathbf{L}_t & \mathbf{L}_t^2 \end{pmatrix}. \quad (3.16)$$

A pontuação r é calculada como

$$r = \det(\mu_{\text{st}}) - k(\text{trace}(\mu_{\text{st}}))^3 = \lambda_1 \lambda_2 \lambda_3 - k(\lambda_1 + \lambda_2 + \lambda_3)^3. \quad (3.17)$$

Definindo as razões como $\alpha = \lambda_2/\lambda_1$ e $\beta = \lambda_3/\lambda_1$, pode-se reescrever a Equação 3.17 como

$$r = \lambda_1^3(\alpha\beta - k(1 + \alpha + \beta)^3). \quad (3.18)$$

A partir da condição $r \geq 0$, k assume seu valor máximo, as razões α e β tendem a 1. Portanto, os STIPs serão encontrados detectando máximos positivos locais em r .



(a) Pontos de interesse no espaço



(b) Pontos de interesse no espaço-tempo

Figura 3.7: Comparação entre a obtenção dos pontos de interesse espaciais e espaço-temporais em uma sequência de vídeo, retirado de [12]

3.3 Pontos de Interesse no Espaço-Tempo com Adaptação de Velocidade e Escala

Em Laptev [29] é proposta uma extensão do STIP clássico, com uma representação STIP que seja estável em situações em que haja variações de escala, velocidade ou ambos.

3.3.1 Adaptação de Escala

Para realizar uma seleção de escala de um evento no espaço-tempo, é preciso definir um operador diferencial que possua variações bruscas tanto na escala temporal quanto na escala espacial [13].

Para a análise, utiliza-se uma pequena região espaço-temporal Gaussiano com variância espacial σ_0^2 e variância temporal τ_0^2

$$\mathbf{I}(x, y, t; \sigma_0^2, \tau_0^2) = \frac{1}{\sqrt{(2\pi)^3 \sigma_l^4 \tau_l^2}} \times \exp(-(x^2 + y^2)/2\sigma_0^2 - t^2/2\tau_0^2) \quad (3.19)$$

onde a representação escala-espaço de \mathbf{I} se dá por

$$\begin{aligned} \mathbf{L}(x, y, t; \sigma^2, \tau^2) &= \mathbf{G}(x, y, t; \sigma^2, \tau^2) * \mathbf{I}(x, y, t; \sigma_0^2, \tau_0^2) \\ &= \mathbf{G}(x, y, t; \sigma_0^2 + \sigma^2, \tau_0^2 + \tau^2). \end{aligned} \quad (3.20)$$

Para recuperar a extensão espaço-temporal de \mathbf{I} , consideram-se as derivadas de segunda ordem normalizadas de \mathbf{L} descritas na Equação 3.21

$$\begin{aligned}\mathbf{L}_{xx,\text{norm}} &= \sigma^{2a}\tau^{2b}\mathbf{L}_{xx}, \\ \mathbf{L}_{yy,\text{norm}} &= \sigma^{2a}\tau^{2b}\mathbf{L}_{yy}, \\ \mathbf{L}_{tt,\text{norm}} &= \sigma^{2c}\tau^{2d}\mathbf{L}_{tt}.\end{aligned}\tag{3.21}$$

Os parâmetros a, b, c e d devem ser determinados de modo que $\mathbf{L}_{xx,\text{norm}}, \mathbf{L}_{yy,\text{norm}}$ e $\mathbf{L}_{tt,\text{norm}}$, com escalas locais $\tilde{\sigma}^2$ e $\tilde{\tau}^2$, possuam escalas extremas em $\tilde{\sigma}^2 = \sigma_0^2$ e $\tilde{\tau}^2 = \tau_0^2$. Para tal, é necessário que as expressões da Equação 3.21 sejam diferenciadas com esses parâmetros.

$$\frac{\partial}{\partial \sigma^2}[\mathbf{L}_{xx,\text{norm}}(0, 0, 0; \sigma^2, \tau^2)] = -\frac{a\sigma^2 - 2\sigma^2 + a\sigma_0^2}{\sqrt{(2\pi)^3(\sigma_0^2 + \sigma^2)^6(\tau_0^2 + \tau^2)}}\sigma^{2(a-1)}\tau^{2b}.\tag{3.22}$$

$$\frac{\partial}{\partial \tau^2}[\mathbf{L}_{xx,\text{norm}}(0, 0, 0; \sigma^2, \tau^2)] = -\frac{2b\tau_0^2 + 2b\tau^2 - \tau^2}{\sqrt{2^5\pi^3(\sigma_0^2 + \sigma^2)^4(\tau_0^2 + \tau^2)^3}}\tau^{2(b-1)}\sigma^{2a}.\tag{3.23}$$

Igualando as Equações 3.22 e 3.23 a zero, obtêm-se as relações dos parâmetros a e b

$$\begin{aligned}a\sigma^2 - 2\sigma^2 + a\sigma_0^2 &= 0, \\ 2b\tau_0^2 + 2b\tau^2 - \tau^2 &= 0.\end{aligned}\tag{3.24}$$

Substituindo $\sigma^2 = \sigma_0^2$ e $\tau^2 = \tau_0^2$, tem-se $a = 1$ e $b = 1/4$. O mesmo processo é feito para a derivada temporal de segunda ordem,

$$\frac{\partial}{\partial \sigma^2}[\mathbf{L}_{tt,\text{norm}}(0, 0, 0; \sigma^2, \tau^2)] = -\frac{c\sigma^2 - \sigma^2 + c\sigma_0^2}{\sqrt{(2\pi)^3(\sigma_0^2 + \sigma^2)^4(\tau_0^2 + \tau^2)^3}}\sigma^{2(c-1)}\tau^{2d}.\tag{3.25}$$

$$\frac{\partial}{\partial \tau^2}[\mathbf{L}_{tt,\text{norm}}(0, 0, 0; \sigma^2, \tau^2)] = -\frac{2d\tau_0^2 + 2d\tau^2 - 3\tau^2}{\sqrt{2^5\pi^3(\sigma_0^2 + \sigma^2)^2(\tau_0^2 + \tau^2)^5}}\tau^{2(d-1)}\sigma^{2c}.\tag{3.26}$$

Gerando as expressões

$$\begin{aligned} c\sigma^2 - 2\sigma^2 + c\sigma_0^2 &= 0, \\ 2d\tau_0^2 + 2d\tau^2 - \tau^2 &= 0. \end{aligned} \quad (3.27)$$

Que após a substituição, resulta em $c = 1/2$ e $d = 3/4$.

A normalização das derivadas da Equação 3.21 garante que todas as derivadas parciais assumem um extremo espaço-escala-tempo no centro da pequena região \mathbf{I} e nas escalas correspondentes às extensões espaciais e temporais de \mathbf{I} . A partir da soma das derivadas parciais, define-se na Equação 3.25 um operador Laplaciano normalizado espaço-temporal.

$$\begin{aligned} \nabla_{norm}^2 \mathbf{L} &= \mathbf{L}_{xx,norm} + \mathbf{L}_{yy,norm} + \mathbf{L}_{tt,norm} \\ &= \sigma^2 \tau^{1/2} (\mathbf{L}_{xx} + \mathbf{L}_{yy}) + \sigma \tau^{3/2} \mathbf{L}_{tt}. \end{aligned} \quad (3.28)$$

Para detectar STIPs com adaptação de escala, os pontos de interesse são tanto máximos no espaço-tempo através da Equação 3.17 quanto no operador normalizado espaço-temporal Laplaciano da Equação 3.28. A Figura 3.8 mostra a detecção de STIPs com adaptação de escala em uma sequência de vídeo.



Figura 3.8: STIP com adaptação de escala, retirado de [13].

3.3.2 Adaptação de Velocidade

A formulação da Equação 3.17 em termos de autovalores implica invariância em relação às rotações 3D do padrão de imagem \mathbf{I} . O domínio do tempo é altamente afetado por Transformadas de Galileu, devido ao movimento constante entre a câmera e o padrão observado [48].

De acordo com Laptev [29], uma Transformada de Galileu \mathbf{G} , mostrada na Equação 3.29, é definida por um vetor de velocidade $(v_x, v_y)^T$ e corresponde a uma transformação linear de coordenadas $p' = \mathbf{G}p$ que possui um efeito de distorção na função da imagem

$\mathbf{L}'(p'; \Sigma') = \mathbf{L}(p; \Sigma)$, onde Σ é uma matriz de covariância 3×3 .

$$\mathbf{G} = \begin{pmatrix} 1 & 0 & v_x \\ 0 & 1 & v_y \\ 0 & 0 & 1 \end{pmatrix}. \quad (3.29)$$

A matriz de covariância Σ da máscara do filtro \mathbf{G}_{st} (Equação 3.15) transforma sob \mathbf{G} como $\Sigma' = \mathbf{G}\Sigma\mathbf{G}^T$, enquanto o gradiente espaço-temporal transforma como $\nabla\mathbf{L}' = \mathbf{G}^{-T}\nabla\mathbf{L}$.

A Equação 3.16 pode ser reescrita como

$$\mu_{st}(x, y, t; s\Sigma) = \mathbf{G}(x, y, t; s\Sigma) * (\nabla\mathbf{L}(\nabla\mathbf{L})^T) = \begin{pmatrix} \mu_{st11} & \mu_{st12} & \mu_{st13} \\ \mu_{st12} & \mu_{st22} & \mu_{st23} \\ \mu_{st13} & \mu_{st23} & \mu_{st33} \end{pmatrix} \quad (3.30)$$

e a transformação de μ_{st} será

$$\mu_{st}'(p'; \Sigma') = \mathbf{G}^{-T} \mu_{st}(p; \Sigma) \mathbf{G}^{-1}. \quad (3.31)$$

A partir da Equação 3.31, nota-se que μ_{st} não é preservado, sendo necessário o cancelamento do efeito da Transformada de Galileu e uma redefinição do operador de interesse em termos de um descritor com adaptação de velocidade $\mu_{st}'' = \mathbf{G}^T \mu_{st}' \mathbf{G}$.

Para estimar \mathbf{G} dos dados, utiliza-se

$$\begin{pmatrix} \mu_{st11}' & \mu_{st12}' \\ \mu_{st12}' & \mu_{st22}' \end{pmatrix} \begin{pmatrix} \tilde{v}_x \\ \tilde{v}_y \end{pmatrix} = \begin{pmatrix} \mu_{st13}' \\ \mu_{st23}' \end{pmatrix}. \quad (3.32)$$

Notando que a estrutura de $(\tilde{v}_x, \tilde{v}_y)^T$ é similar ao cálculo de fluxo ótico proposto por Lucas [49]. Por conseguinte, usa-se a matriz μ_{st}' para determinar as velocidades, obtendo

$$\tilde{v}_x = \frac{\mu_{st22}' \mu_{st13}' - \mu_{st12}' \mu_{st23}'}{\mu_{st11}' \mu_{st22}' - \mu_{st12}'^2}, \tilde{v}_y = \frac{\mu_{st11}' \mu_{st23}' - \mu_{st12}' \mu_{st13}'}{\mu_{st11}' \mu_{st22}' - \mu_{st12}'^2}. \quad (3.33)$$

Com as velocidades definidas, um descritor μ_{st}'' de velocidade adaptada pode ser utilizado. Logo, para qualquer Transformada de Galileu inicial $\mathbf{G}(v_x, v_y)$, $\mu_{st}'' = \mathbf{G}^T(\tilde{v}_x, \tilde{v}_y) \mu_{st}' \mathbf{G}(\tilde{v}_x, \tilde{v}_y)$ terá uma diagonal com elementos $\mu_{st13}'' = \mu_{st23}'' = 0$. Esta diagonal servirá para computar os pontos independentemente da Transformada de Galileu.

A Equação 3.15 com adaptação de velocidade pode ser redefinida como

$$r_{vel} = \det(\mu_{st}'') - k(\text{trace}(\mu_{st}''))^3. \quad (3.34)$$

A metodologia proposta apresentada no Capítulo 4 é fundamentada na modelagem matemática apresentada neste Capítulo e será utilizada para a construção dos descritores espaço-temporais.

Capítulo 4

Metodologia

A metodologia proposta neste Capítulo tem como objetivo o reconhecimento e classificação de ações humanas em sequências de vídeo e baseia-se nos trabalhos de Laptev [17], [29]. Para facilitação da leitura, a nomenclatura dada a cada um destes é C-STIP e V-STIP, respectivamente referindo-se ao método de STIP Clássico e STIP com Adaptação de Velocidade e Escala, seguindo os conceitos apresentados no Capítulo 3.

Usualmente, as técnicas de STIP são utilizadas como detectores locais. Para o processo de reconhecimento e classificação, na maioria dos trabalhos encontrados na literatura que fazem uso desta técnica, é necessário o uso de descritores adicionais, como Histogramas de Gradientes Orientados (HOG) [27] ou fluxo ótico [39], para que as características locais sejam melhoradas. Neste trabalho, o STIP será usado como descritor principal e único, sem o auxílio de nenhum outro descritor auxiliar, sendo esta uma abordagem de análise da metodologia proposta.

O fluxograma simplificado da Figura 4.1 mostra as etapas básicas da metodologia, que serão apresentadas em Seções separadas a seguir.

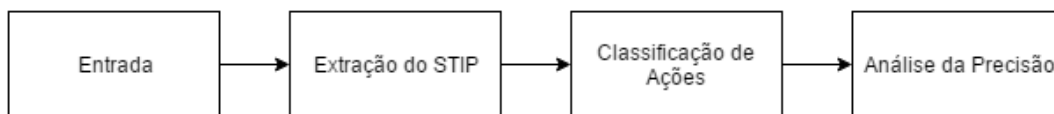








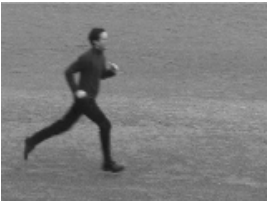









Figura 4.1: Fluxograma da metodologia proposta.

4.1 Entrada

Para avaliação completa da metodologia, bases de vídeos públicos especializadas em reconhecimento de ações humanas foram usados. As principais bases de dados foram: KTH [30], UCF101 [50], Weizmann [32], YouTube [51]. Uma amostra de vídeos das bases utilizadas e de cada classe usada neste trabalho, podem ser encontrados na Tabela 4.1 e maiores detalhes são apresentados nas Subseções a seguir.

Tabela 4.1: Amostras das bases de dados.

KTH	UCF101	Weizmann	YouTube
			
<i>Boxing</i>	<i>Biking</i>	<i>Bend</i>	<i>Basketball</i>
			
<i>Handwaving</i>	<i>Jumping Jack</i>	<i>Gallop Sideways</i>	<i>Diving</i>
			
<i>Running</i>	<i>Punch</i>	<i>Jump in Place</i>	<i>Soccer Juggling</i>
			
<i>Walking</i>	<i>Walking with Dog</i>	<i>Skip</i>	<i>Volleyball Spiking</i>

4.1.1 KTH

Com seis diferentes tipos de ações humanas executados diversas vezes por 25 objetos em quatro cenários diferentes (ao ar livre, ao ar livre com variação de escala, ao ar livre com roupas diferentes e ambiente fechado), a base KTH [30] contém 2391 sequências com plano de fundo homogêneo.

Os vídeos possuem uma resolução de 160×120 pixels 25 quadros por segundo (*Frames per Second* – FPS) e as ações usadas neste trabalho, com vídeos de 20 segundos em média e dez vídeos por ação, foram: *boxing*, *handwaving*, *running* e *walking*.

Os vídeos da ação *boxing* apresentam um indivíduo lateralmente posicionado no centro da câmera movimentando apenas os dois braços, verticalmente e com extensão completa dos cotovelos. A ação *handwaving* mostra um sujeito posicionado frontalmente no centro da câmera, movimentando os dois braços em forma de arco.

Para as ações *walking* e *running*, os indivíduos cruzam o campo de visão da câmera de maneira similar, entretanto, mais lentamente para a ação *walking* e com velocidade elevada

para *running*.

4.1.2 UCF101

A base de dados UCF101 [50] possui 13320 vídeos e 101 ações, com uma grande variedade de variações no movimento da câmera, aparição de objetos, escala de objetos, ponto de vista, mudanças de iluminação, entre outros. Essas variações tornam a base complexa, visto que a maioria das outras bases trazem cenários controlados.

Todos os vídeos possuem resolução e cadência fixas de 25 FPS e 320×240 *pixels*, respectivamente. Neste trabalho, dez vídeos por ação com duração média de cinco segundos cada foram usados e as seguintes classes de ações foram selecionadas: *biking*, *jumping jack*, *punch* e *walking with dog*.

A classe *biking* apresenta entre um a dois indivíduos andando de bicicleta em ambientes abertos. Os cenários são dinâmicos e diferentes para cada vídeo, com montanhas, campos, árvores ou centros urbanos. O mesmo ocorre na ação *walking with dog*, onde um humano e um animal caminham juntos em cenários dinâmicos.

Para a classe *punch*, os cenários são ringues de lutas, com plateia, juiz e dois lutadores. Nem sempre os lutadores utilizam luvas vermelhas, ocorrendo o uso das mãos livres em alguns vídeos.

A ação *jumping jack* contém vídeos de pessoas fazendo polichinelos tanto ao ar livre quanto em ambientes fechados. Todos se encontram de frente para a câmera, variando, entretanto, seu posicionamento.

4.1.3 Weizmann

Em Blank [32], uma base de dados com 90 sequências de vídeo de resolução 180×144 *pixels* e cadência 50 FPS é criada.

Esta base contém dez ações realizadas por nove objetos distintos e plano de fundo estático. Para a metodologia proposta, quatro dessas ações foram destacadas, sendo elas: *bend*, *gallop sideways*, *jump in place* e *skip*. A duração média de cada vídeo das ações escolhidas é de dois segundos.

A classe *bend* mostra uma pessoa abaixando lateralmente, mas estática. Para *gallop sideways*, há movimentação horizontal enquanto uma pessoa pula abrindo e fechando as pernas. A ação *jump in place* apresenta um indivíduo pulando com braços e pernas fechados no centro da câmera. A classe *skip* mostra uma pessoa cruzando o cenário enquanto salta com uma das pernas flexionadas por todo o vídeo.

4.1.4 YouTube

A base de dados YouTube [51] abrange 11 ações realistas com totalidade de 1600 vídeos, capturadas do *site* de compartilhamento de vídeos YouTube. Assim como a base UCF101, também é considerada uma base complexa, pois possui variações de iluminação, escala de objetos, oclusões e outras características sobreditas.

Aqui, optou-se pelo uso das classes: *basketball*, *diving*, *soccer juggling*, *volleyball spiking*; Consistindo em sete vídeos por ação com extensão aproximada de sete segundos. Todos os vídeos se passam em ambientes esportivos, como ginásios, quadras e piscinas. Para a classe *basketball*, um ou mais jogadores lançam bolas na cesta. Na ação *diving*, um indivíduo salta de um trampolim em uma piscina, lateralmente posicionado para a câmera.

A classe *volleyball spiking* contém vídeos com jogadores rebatendo bolas por cima das redes de vôlei, dentro de quadras específicas e a classe *soccer juggling* mostra um ou mais jogadores fazendo embaixadinhas.

Todos os vídeos das classes selecionadas de cada base de dados foram separados em conjuntos de treinamento (70%) e conjunto de validação (30%). Esta abordagem foi escolhida para melhor avaliar a proposta, sendo que assim além de padronizar as quantidades de vídeos para classificação, permite avaliar a real capacidade de classificação destes descritores sem o auxílio de outros tipos de descritores.

A escolha das ações de cada classe foi realizada propositalmente, de modo que a comparação entre as técnicas de C-STIP e V-STIP sejam metodologicamente adequadas. Entretanto, o número reduzido de classes se deu devido à grande e custosa carga computacional que cada vídeo processado demandava.

4.2 Extração dos Descritores STIP

Os algoritmos de extração do C-STIP e V-STIP foram baseados nas abordagens estado-da-arte, estas apresentadas no Capítulo 3. Os vetores de saída de ambas as técnicas serão apresentados na Seção 4.3.

4.2.1 C-STIP

A entrada do algoritmo consiste em uma sequência de imagens extraídas dos vídeos provenientes das bases de vídeos utilizadas, o parâmetro k da Equação 3.17 e valores inteiros para as variações temporais e espaciais.

O cálculo da representação linear é feito de maneira que o sinal 3D \mathbf{I} seja convoluído com uma máscara Gaussiana espacial com variância σ_l^2 e uma máscara Gaussiana temporal com variância τ_l^2 . Após essa etapa, as derivadas de primeira-ordem podem ser obtidas.

Para a computação de μ_{st} , a transformação mostrada na Equação 3.30 é considerada, de forma que os termos $\mu_{st_{11}}, \mu_{st_{12}}, \mu_{st_{13}}, \mu_{st_{22}}, \mu_{st_{23}}, \mu_{st_{33}}$ são obtidos separadamente, facilitando os dois estágios subsequentes, determinante e função traço da matriz μ_{st} .

Com o resultado da função Harris (Capítulo 3), os máximos locais espaço-temporais são apurados, resultando em um vetor de posições e um vetor de valores respectivos às posições, onde os valores são os índices lineares de cada elemento diferente de zero em (x, y, z) .

O vetor de pontos contém a posição dos pontos, as velocidades estimadas e os termos de μ citados anteriormente. Os pontos com maior *score* desse vetor de pontos são organizados de acordo com os valores obtidos na etapa de máximos locais, classificados do maior valor para o menor, gerando uma nova variável de índice.

Por fim, gerando a matriz de saída *cstip*, os pontos são reorganizados de forma que os mais fortes estejam nas linhas superiores. Todos os passos de implementação são descritos no Algoritmo 1.

Algoritmo 1: Algoritmo proposto para C-STIP

Entrada: $[I(x, y, t), \sigma_l^2, \sigma_i^2, \tau_l^2, \tau_i^2, k]$

Saída: [*cstip*]

1 **início**

2 | Cálculo da representação linear **L** (Eq. 3.14);

3 | Cálculo das derivadas de primeira-ordem L_x, L_y, L_t ;

4 | Cálculo de μ_{st} (Eq. 3.16);

5 | Cálculo da determinante de μ_{st} ;

6 | Cálculo do traço de μ_{st} ;

7 | Cálculo de r (Eq. 3.17);

| /* pos = posição, val = valor */

8 | [*pos, val*] = máximo local;

9 | **se** tamanho(*pos*) > 0 **então**

10 | | Estimação das velocidades v_x, v_y ;

| /* pts = pontos STIP */

11 | | Criação do vetor de pontos [*pts*];

12 | **fim**

13 | Seleção dos pontos mais fortes;

14 | Criação da matriz de saída [*cstip*];

15 **fim**

4.2.2 V-STIP

A primeira parte do algoritmo implementado para o V-STIP é construída de maneira semelhante ao algoritmo do C-STIP, apresentado anteriormente.

A partir dos pontos iniciais P_i encontrados por C-STIP, uma busca iterativa é realizada de modo que um ponto é procurado em **I** na vizinhança do ponto inicial que deve possuir as seguintes características: I) é um máximo local de uma função de Harris e II) é um extremo

de operador Laplaciano normalizado nas escalas (σ^2, τ^2) .

As posições dos pontos de interesse e as escalas são atualizadas ao selecionar a escala espaço-temporal vizinha que maximiza $(\nabla_{norm}^2 \mathbf{L})^2$ e então recalculando a função de Harris com as novas escalas.

O cálculo da adaptação de escalas respeita as Equações 3.33 a 3.34. A matriz de saída *vstip* possui mais características do que a matriz *costip* devido ao acréscimo das escalas, descritas no Algoritmo 2.

4.2.3 Análise da Complexidade dos Algoritmos

A complexidade é uma forma de analisar os algoritmos de modo que se possa prever os recursos que ele necessitará em sua execução [52]. Realizando uma breve análise da complexidade dos pseudo códigos apresentados acima, nota-se que para o primeiro algoritmo temos uma complexidade $\Theta(1)$, por se tratar de um código direto e sem laços que dependam de variáveis externas. Para o segundo algoritmo, consideramos o número máximo de iterações como o pior caso. Dado que este número é representado pela quantidade n de quadros existentes no vídeo, a complexidade é, então, $\Theta(n)$.

4.3 Classificação das Ações

Três metodologias de classificação foram estipuladas para este trabalho, SVMs, Redes Neurais de Reconhecimento de Padrões e Redes Neurais de Ajustes de Funções. A escolha destes métodos de baixa complexidade deu-se pela intenção da avaliação do desempenho dos descritores STIP sem o auxílio de descritores externos, sendo ideal para apurar os resultados atingidos.

4.3.1 Máquinas de Vetores de Suporte

As máquinas de vetores de suporte (SVM) [53] são classificadores discriminativos definidos por um hiperplano de separação, que, a partir de um conjunto de dados de treinamento, produz um hiperplano ótimo que engloba novos exemplos. Isto é, o SVM produz um modelo que é capaz de prever a classe de um dado de validação baseando-se apenas nas características dadas no conjunto de treinamento.

Neste trabalho, o algoritmo *Support Vector Classification* (SVC) [54] com um núcleo de base radial (Equação 4.1), é usado para treinar e prever as classes do conjunto de validação.

$$K(l, h) = \exp(-\gamma \|l - h\|^2) \quad (4.1)$$

Algoritmo 2: Algoritmo proposto para V-STIP

Entrada: $[I(x, y, t), \sigma_l^2, \sigma_i^2, \tau_l^2, \tau_i^2, k]$ **Saída:** $[vstip]$ 1 **início**2 Cálculo da representação linear \mathbf{L} (Eq. 3.14);3 Cálculo das derivadas de primeira-ordem L_x, L_y, L_t ;4 Cálculo de μ_{st} (Eq. 3.16);5 Cálculo da determinante de μ_{st} ;6 Cálculo do traço de μ_{st} ;7 Cálculo de r (Eq. 3.17);

/* pos = posição, val = valor */

8 $[pos, val] = \text{máximo local}$;9 **se** $tamanho(pos) > 0$ **então**10 | Estimacão das velocidades v_x, v_y ;

| /* pts = pontos STIP */

11 | Criacão do vetor de pontos $[pts]$;12 **fim**

13 Seleção dos pontos mais fortes;

/* $[cstipvel]$ é a matriz $[cstip]$ com velocidade */14 Criacão da matriz de saída $[cstipvel]$;/* $P_i = \text{ponto de interesse inicial}$ */15 **para cada** P_i **faça**16 | **enquanto** *houver convergência dos valores de parâmetro ou atingir o número máximo de iterações* **faça**17 | | Computacão de $\nabla_{norm}^2 \mathbf{L}$ (Eq.3.28) em (x, y, t) e combinações de escalas vizinhas $(\tilde{\sigma}^2, \tilde{\tau}^2)$;18 | | Escolha da combinaçã de escalas de integraçã que maximizam $(\nabla_{norm}^2 \mathbf{L})^2$;19 | | **se** *nova escolha for melhor* **então**

20 | | | Atualizaçã de escalas

21 | | | **senão**

22 | | | Desconsiderar a etapa

23 | | | **fim**24 | | Cálculo de r (Eq. 3.17) com as novas escalas;25 | | **se** *Encontrar pontos Harris* **então**26 | | | Atualizaçã do vetor de posiçã de acordo com o *warp* de velocidade;27 | | | Conversã das posições para coordenadas (x, y, t) ;

28 | | | Atualizaçã da velocidade;

29 | | | Seleçã do ponto mais próximo;

30 | | | Atualizaçã do ponto de interesse;

31 | | | **fim**32 | | **fim**33 | **fim**34 Criacão da matriz de saída $[vstip]$;35 **fim**

Onde γ é o parâmetro do núcleo.

4.3.2 Redes Neurais Artificiais

As Redes Neurais Artificiais (RNAs) são métodos matemáticos e computacionais de simulação da funcionalidade do cérebro humano [14]. A partir de algoritmos de aprendizagem, uma estrutura similar ao cérebro processa funcionalidades que remetem à maneira que o conhecimento humano é adquirido, por meio de experiências [14].

Para reproduzir o cérebro, é necessário construir neurônios. O perceptron é um modelo de neurônio baseado em McCulloch [55], possuindo um conjunto de ligações, ou sinapses, artifício necessário para que os algoritmos de aprendizagem tenham êxito.

Dois modelos de RNAs são utilizados para treinar as classes selecionadas nesta metodologia proposta, sendo elas: rede para reconhecimento de padrões e rede de ajuste de funções.

Ambos os modelos possuem a arquitetura de redes alimentadas adiante com múltiplas camadas, onde a primeira camada possui uma conexão com a entrada da rede e as camadas subsequentes têm conexões com as suas respectivas camadas anteriores. A última camada produz a saída da rede [56].

As redes de reconhecimento de padrões e de ajuste de funções são versões especializadas das redes alimentadas adiante, sendo a forma final de saída o que diferencia uma da outra. O Anexo 6.3 possui informações mais detalhadas sobre ambos os modelos de RNAs utilizadas na classificação.

O algoritmo de treinamento escolhido para as redes neurais foi o algoritmo de escala conjugada, baseado em direções conjugadas, devido à sua capacidade de utilizar menos memória para fazer esta operação [57].

4.3.3 Conjuntos de Dados

O conjunto de dados utilizado corresponde à saída dos algoritmos apresentados anteriormente. Com a base de dados previamente separada em treinamento e validação, o conjunto de treinamento de C-STIP é dado por $(a_{c_i}, b_i), i = 1, \dots, l$, onde $a_{c_i} \in \mathfrak{R}$ é uma matriz de características e $b_i \in \{1, 2, 3, 4\}$.

A matriz de características $[cstip]$, por sua vez, possui uma dimensão $11 \times n$, onde n é a quantidade de pontos encontrados em um vídeo. As onze colunas representam as posições dos pontos x, y, z , as velocidades v_x e v_y e $\mu_{st_{11}}, \mu_{st_{12}}, \mu_{st_{13}}, \mu_{st_{22}}, \mu_{st_{23}}, \mu_{st_{33}}$. As características referentes às escalas foram retiradas por terem alta correlação e, portanto, serem dispensáveis no processo de classificação. A matriz completa e a entrada do SVM, x_{c_i} , é a concatenação entre a matriz de características de todos os vídeos presentes no treinamento da base de dados.

O conjunto de treinamento de V-STIP é dado por $(a_{v_i}, b_i), i = 1, \dots, l$, onde $a_{v_i} \in \mathfrak{R}$ é uma matriz de características e $b_i \in \{1, 2, 3, 4\}$. A matriz $[vstip]$ tem o tamanho $13 \times n$, onde n é a quantidade de pontos encontrados em um vídeo. Suas colunas são: as posições dos pontos x, y, z , as escalas σ^2 e τ^2 , as velocidades v_x e v_y e $\mu_{st_{11}}, \mu_{st_{12}}, \mu_{st_{13}}, \mu_{st_{22}}, \mu_{st_{23}}, \mu_{st_{33}}$. Da mesma maneira que o C-STIP, a matriz de entrada x_{v_i} corresponde à junção da matriz $[vstip]$ em cada vídeo.

Ressalta-se que os conjuntos de validação possuem a mesma estrutura dos conjuntos de treinamento, porém, não possuem identificação da classe pertencente.

4.4 Análise da Precisão

As métricas de avaliação comumente utilizadas para verificar a qualidade dos resultados obtidos em um experimento de aprendizado de máquina são precisão, revocação e medida-F [58].

Medidas binárias são definidas para calcular essas métricas, sendo elas: o verdadeiro positivo (*true positive* – tp), que representa um item sendo corretamente considerado como relevante; falso positivo (*false positive* – fp), quando um item é erroneamente computado como relevante; verdadeiro negativo (*true negative* – tn), representando um objeto corretamente considerado como irrelevante e falso negativo (*false negative* – fn), onde um item é erroneamente classificado como irrelevante.

A revocação, também conhecida pelo termo em inglês *recall*, representa a proporção entre casos positivos que foram corretamente previstos como tal, entretanto não costuma ser usada como métrica de avaliação por não dar nenhum rastro sobre os itens que não foram retornados [58]. A Equação 4.2 define a revocação como:

$$recall = \frac{tp}{tp + fn}. \quad (4.2)$$

A precisão, também conhecida pelo termo em inglês *precision*, denota a proporção entre os casos previstos como positivos e que são realmente positivos. Desta maneira, a precisão é a métrica mais utilizada para a avaliação de resultados [59] e é representada pela Equação 4.3.

Além do uso da precisão, diversos autores [26, 11] representam a análise dos dados com uma métrica derivada da precisão, a precisão média (*Mean Average Precision* – MAP). Esta métrica corresponde à média do somatório de todas as precisões em um dado conjunto de dados e está descrita na Equação 4.4.

$$precision = \frac{tp}{tp + fp}. \quad (4.3)$$

$$MAP = \frac{1}{n} \sum_{i=1}^n precision_i. \quad (4.4)$$

Uma média harmônica entre essas duas medidas gera a medida-F (*F-measure*), apresentada na Equação 4.5, contudo, da mesma maneira que os teste anteriores, a medida-F também não considera os itens negativos que foram corretamente classificados como negativos.

$$fmeasure = 2 \cdot \frac{precision \cdot recall}{precision + recall}. \quad (4.5)$$

A acurácia possui o escopo completo de informações sobre os dados , tanto positivas, os acertos, quanto negativas, os erros, e é dada pela Equação 4.6, onde *acc* é a acurácia.

$$acc = \frac{tp + tn}{tp + tn + fp + fn}. \quad (4.6)$$

Os dados classificados, particularmente para o caso da metodologia aqui proposta, não possuem verdadeiros negativos, sendo assim, as medidas clássicas de precisão e medida-F não interferem na avaliação dos resultados. Seguindo trabalhos mais recentes e relacionados como os de Chakraborty [11] e Dehghan [26], a precisão média (MAP) foi selecionada como métrica principal de avaliação, sendo utilizada para todas as classes e bases de dados.

A metodologia proposta neste Capítulo será avaliada e discutida no Capítulo 5, sendo esta realizada a partir de gráficos e tabelas detalhadas contendo os resultados alcançados para os diversos cenários de experimentos.

Capítulo 5

Resultados

Neste Capítulo são apresentados os resultados obtidos a partir da implementação da metodologia e dos algoritmos propostos no Capítulo 4, com as devidas referências técnicas no Capítulo 3, bem como as especificações técnicas do ferramental computacional utilizado nos testes. Ressalta-se que a validação aqui utilizada foi considerada para classificadores de menor complexidade, como o SVM e RNAs, visto que dada a hipótese do uso de STIPs sem descritores auxiliares e tendo bons resultados com classificadores elementares, em que aumentando o grau de complexidade destes, os resultados também sofreriam, consequentemente, uma melhoria no desempenho alcançado.

O detalhamento das máquinas usadas para executar metodologia proposta seguem as configurações de *hardware* descritas na Tabela 5.1. Múltiplos computadores foram utilizados para que o processamento ocorresse do modo mais breve possível.

Tabela 5.1: Configurações utilizadas para processamento dos dados.

Máquina	Memória	Processador	Sistema Operacional
1	4 GB	Intel i3 2.20 GHz 4 núcleos	Linux Mint 17.3 'Rosa'
2	8 GB	Intel i7 3.20 GHz 8 núcleos	Linux Mint 17.2 'Rafaela'
3	16 GB	Intel i7 3.20 GHz 8 núcleos	Windows 10

Como apresentado anteriormente no Capítulo 4, os conjuntos de treinamento e validação são divididos em 70% – 30%, respectivamente. Ademais, para reforçar os resultados obtidos, as curvas de Característica de Operação do Receptor (*Receiver Operating Characteristic* – ROC), que representa o desempenho do treinamento da rede neural, serão também apresentadas neste Capítulo.

Para o treinamento do classificador SVM, implementado a partir da biblioteca *LIBSVM* [60], os parâmetros de entrada foram variados de modo que a melhor performance pudesse ser atingida. Os parâmetros selecionados foram: O fator de sensibilidade da função de Harris k e as variações espaciais σ_l^2 e temporais τ_l^2 , todos descritos na Equação 3.17.

Para ambas as redes neurais, as mesmas alterações de parâmetros foram feitas, entretanto, o algoritmo de treinamento usado foi baseado em direções conjugadas e implementado pela ferramenta MATLAB [56] a partir da *toolbox* de redes neurais artificiais.

Para avaliação da metodologia, elaborou-se sete cenários de treinamento (CT) para cada implementação, C-STIP e V-STIP respectivamente, encontrados na Tabela 5.2.

Tabela 5.2: Cenários de Treinamento.

Cenário de Treinamento	σ_l^2	τ_l^2	k
CT_1	4	2	0.01
CT_2	4	2	0.05
CT_3	4	2	0.1
CT_4	4	2	0.3
CT_5	9	3	0.05
CT_6	16	4	0.05
CT_7	25	5	0.05
CT_8	36	6	0.05

O oitavo cenário, CT_8 , chegou a ser testado, porém os resultados de classificação obtiveram problemas de convergência, sendo, então, retirado como cenário de treinamento. Destes cenários, após avaliação do classificador menos complexo SVM, um conjunto de subcenários contendo os melhores resultados foi proposto para o treinamento e validação das redes neurais. Os cenários selecionados foram: CT_1 , CT_2 , CT_5 e CT_6 .

5.1 Resultados da Classificação utilizando SVM

Os resultados obtidos com a classificação do SVM foram compilados, para cada base de dados e ação selecionada, em matrizes de confusão encontradas nas Figuras 6.1 a 6.28, no Capítulo de Apêndice 6, onde as diagonais estão em evidência nas Tabelas 5.3 a 5.6. As diagonais principais representam as precisões encontradas.

Tabela 5.3: Diagonais destacadas da base KTH.

Cenário	Base							
1	KTH							
	<i>Boxing</i>		<i>Handwaving</i>		<i>Running</i>		<i>Walking</i>	
	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP	V-STIP
	42%	50%	39%	42%	47%	51%	40%	37%
2	<i>Boxing</i>		<i>Handwaving</i>		<i>Running</i>		<i>Walking</i>	
	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP	V-STIP
	43%	52%	34%	42%	56%	51%	41%	35%
	3	<i>Boxing</i>		<i>Handwaving</i>		<i>Running</i>		<i>Walking</i>
C-STIP		V-STIP	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP	V-STIP
42%		88%	40%	45.5%	47%	48%	40%	35.5%
4		<i>Boxing</i>		<i>Handwaving</i>		<i>Running</i>		<i>Walking</i>
	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP	V-STIP
	48%	58%	27%	34%	51%	55%	49%	38.5%
	5	<i>Boxing</i>		<i>Handwaving</i>		<i>Running</i>		<i>Walking</i>
C-STIP		V-STIP	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP	V-STIP
48%		52%	46%	43%	52%	51%	30%	34%
6		<i>Boxing</i>		<i>Handwaving</i>		<i>Running</i>		<i>Walking</i>
	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP	V-STIP
	50%	62%	47%	44%	61%	46%	30%	37%
	7	<i>Boxing</i>		<i>Handwaving</i>		<i>Running</i>		<i>Walking</i>
C-STIP		V-STIP	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP	V-STIP
52%		76%	47%	42%	68%	93%	27%	38%

Tabela 5.4: Diagonais destacadas da base UCF101.

Cenário	Base							
1	UCF101							
	<i>Biking</i>		<i>Jumping Jack</i>		<i>Punch</i>		<i>Walking With Dog</i>	
	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP	V-STIP
	50%	45%	33%	47%	43%	44%	34%	48%
2	<i>Biking</i>		<i>Jumping Jack</i>		<i>Punch</i>		<i>Walking With Dog</i>	
	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP	V-STIP
	52%	49%	39%	33%	39%	44%	37%	42%
	<i>Biking</i>		<i>Jumping Jack</i>		<i>Punch</i>		<i>Walking With Dog</i>	
3	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP	V-STIP
	47%	43%	45.5%	29.5%	48%	45.5%	36%	40%
	<i>Biking</i>		<i>Jumping Jack</i>		<i>Punch</i>		<i>Walking With Dog</i>	
	4	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP
54%		49%	29%	31%	42%	42%	42%	42%
<i>Biking</i>		<i>Jumping Jack</i>		<i>Punch</i>		<i>Walking With Dog</i>		
5		C-STIP	V-STIP	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP
	65%	57%	34%	27%	49%	53%	43%	49%
	<i>Biking</i>		<i>Jumping Jack</i>		<i>Punch</i>		<i>Walking With Dog</i>	
	6	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP
62%		53%	30%	29%	47%	51%	35%	40%
<i>Biking</i>		<i>Jumping Jack</i>		<i>Punch</i>		<i>Walking With Dog</i>		
7		C-STIP	V-STIP	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP
	52%	26%	29%	30%	39%	39%	46%	65%

Tabela 5.5: Diagonais destacadas da base Weizmann.

Cenário	Base							
1	Weizmann							
	<i>Bend</i>		<i>Jump in Place</i>		<i>Gallop Sideways</i>		<i>Skip</i>	
	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP	V-STIP
	35%	52%	43%	45.5%	37%	53%	41%	34%
2	<i>Bend</i>		<i>Jump in Place</i>		<i>Gallop Sideways</i>		<i>Skip</i>	
	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP	V-STIP
	23%	42%	34%	34%	38%	35.5%	39%	38%
	<i>Bend</i>		<i>Jump in Place</i>		<i>Gallop Sideways</i>		<i>Skip</i>	
3	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP	V-STIP
	23%	41%	39%	42%	35%	33%	36%	33%
	<i>Bend</i>		<i>Jump in Place</i>		<i>Gallop Sideways</i>		<i>Skip</i>	
	4	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP
26%		42%	39%	40.5%	35%	32%	38%	38%
<i>Bend</i>		<i>Jump in Place</i>		<i>Gallop Sideways</i>		<i>Skip</i>		
5		C-STIP	V-STIP	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP
	35%	62%	49%	35%	34%	28%	39%	37%
	<i>Bend</i>		<i>Jump in Place</i>		<i>Gallop Sideways</i>		<i>Skip</i>	
	6	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP
48%		57%	40%	33%	35%	29%	44%	39%
<i>Bend</i>		<i>Jump in Place</i>		<i>Gallop Sideways</i>		<i>Skip</i>		
7		C-STIP	V-STIP	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP
	46%	21%	40%	43%	34%	34%	41%	45%

Tabela 5.6: Diagonais destacadas da base YouTube.

Cenário	Base							
1	YouTube							
	<i>Shooting</i>		<i>Diving</i>		<i>Juggling</i>		<i>Spiking</i>	
	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP	V-STIP
	23%	14%	51%	49%	54%	52%	76%	81%
2	<i>Shooting</i>		<i>Diving</i>		<i>Juggling</i>		<i>Spiking</i>	
	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP	V-STIP
	44%	22%	49%	52%	52%	48%	75%	79%
	3	<i>Shooting</i>		<i>Diving</i>		<i>Juggling</i>		<i>Spiking</i>
C-STIP		V-STIP	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP	V-STIP
46%		23%	51%	53%	31%	31%	85%	86%
4		<i>Shooting</i>		<i>Diving</i>		<i>Juggling</i>		<i>Spiking</i>
	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP	V-STIP
	46.5%	21.5%	53%	54%	51%	52%	78%	81.5%
	5	<i>Shooting</i>		<i>Diving</i>		<i>Juggling</i>		<i>Spiking</i>
C-STIP		V-STIP	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP	V-STIP
63%		20%	51%	53%	52%	49%	81%	82%
6		<i>Shooting</i>		<i>Diving</i>		<i>Juggling</i>		<i>Spiking</i>
	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP	V-STIP
	60%	29%	47%	46%	50%	46%	82%	83%
	7	<i>Shooting</i>		<i>Diving</i>		<i>Juggling</i>		<i>Spiking</i>
C-STIP		V-STIP	C-STIP	V-STIP	C-STIP	V-STIP	C-STIP	V-STIP
25%		24%	47%	46%	52%	45%	73%	90%

Baseado nas matrizes de confusão, as medidas MAP foram calculadas a partir de todas as classes, resumindo os resultados globais. As tabelas 5.7 a 5.13 apresentam o MAP para cada método e os maiores valores de cada método encontram-se destacados.

Tabela 5.7: CT_1 - Mean Average Precision (MAP)

	KTH	UCF101	Weizmann	YouTube
C-STIP	42%	40%	39%	51%
V-STIP	45%	46%	46%	49%

Tabela 5.8: CT_2 - Mean Average Precision (MAP)

	KTH	UCF101	Weizmann	YouTube
C-STIP	42%	44%	33%	53%
V-STIP	54%	39%	37%	48%

Tabela 5.9: CT_3 - Mean Average Precision (MAP)

	KTH	UCF101	Weizmann	YouTube
C-STIP	42%	44%	33%	53%
V-STIP	54%	40%	37%	48%

Tabela 5.10: CT_4 - Mean Average Precision (MAP)

	KTH	UCF101	Weizmann	YouTube
C-STIP	44%	42%	34%	57%
V-STIP	46%	41%	38%	52%

Tabela 5.11: CT_5 - Mean Average Precision (MAP)

	KTH	UCF101	Weizmann	YouTube
C-STIP	44%	47%	39%	61%
V-STIP	45%	46%	40%	51%

Tabela 5.12: CT_6 - Mean Average Precision (MAP)

	KTH	UCF101	Weizmann	YouTube
C-STIP	47%	43%	42%	60%
V-STIP	47%	43%	39%	51%

Tabela 5.13: CT_7 - Mean Average Precision (MAP)

	KTH	UCF101	Weizmann	YouTube
C-STIP	48%	41%	40%	49%
V-STIP	62%	40%	36%	51%

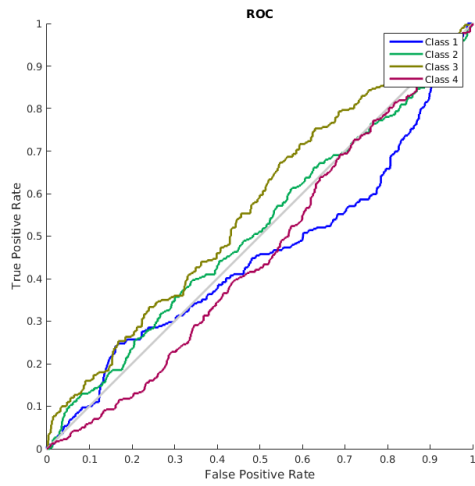
5.2 Resultados da Classificação utilizando RNA

Para os resultados obtidos com o RNA, um estudo exaustivo foi realizado de modo que o melhor número de neurônios e quantidade de camadas fosse encontrado e usado para os resultados finais. A seguir, os resultados dos dois modelos de RNA propostos serão explicitados separadamente. Os resultados completos dos gráficos encontram-se disponíveis no Apêndice 6.

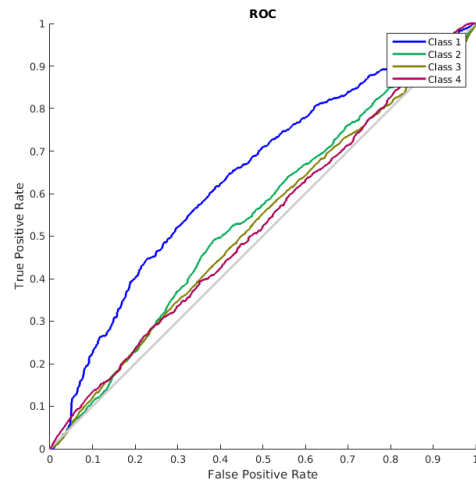
As curvas ROC utilizadas para representar os modelos de redes neurais, consistem em um diagrama que representa a sensibilidade (taxa de verdadeiros positivos ou *true positive rate*) em função da especificidade (taxa de falsos positivos ou *false positive rate*) [61].

5.2.1 Redes Neurais de Reconhecimento de Padrões

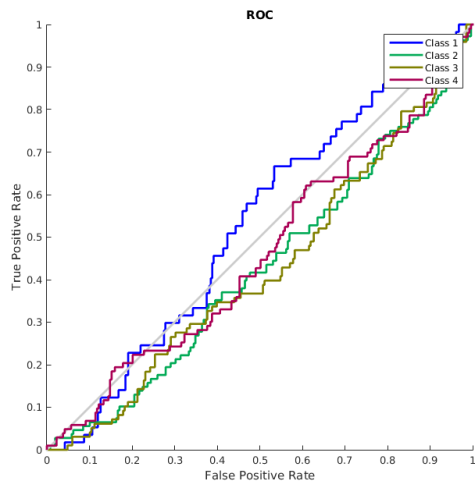
Durante o processo de treinamento e também para a validação e teste, foi feita a análise a partir das curvas ROCs da rede neural de reconhecimento de padrões. Para melhor visualização, apenas o melhor cenário será mostrado a seguir.



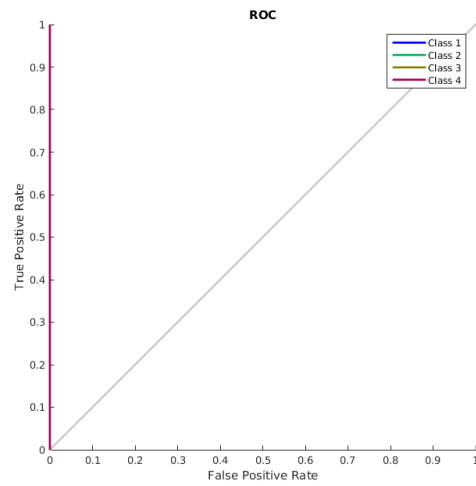
(a) Base de dados KTH



(b) Base de dados UCF101

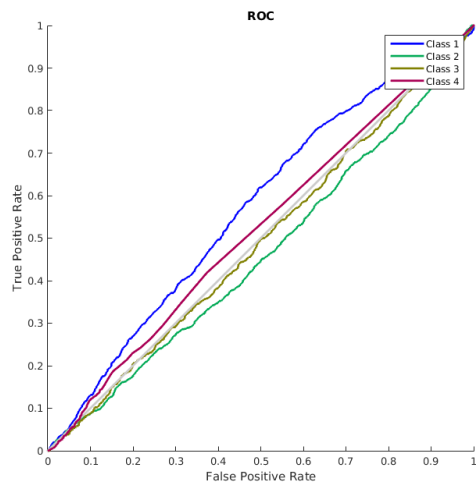


(c) Base de dados Weizmann

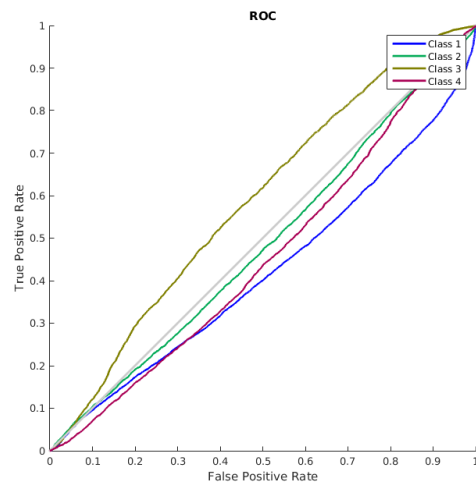


(d) Base de dados YouTube

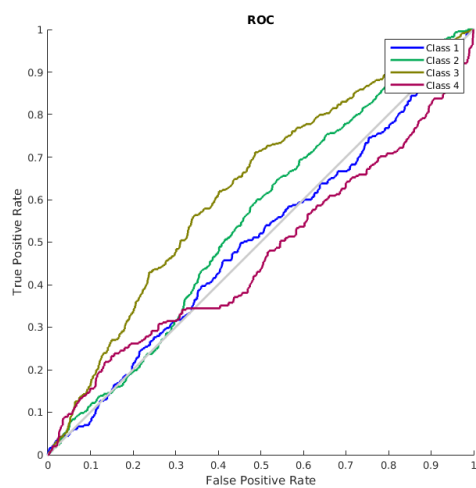
Figura 5.1: Curvas ROC do método C-STIP para o cenário de treinamento CT_6 utilizando a classificação de redes neurais de reconhecimento de padrões.



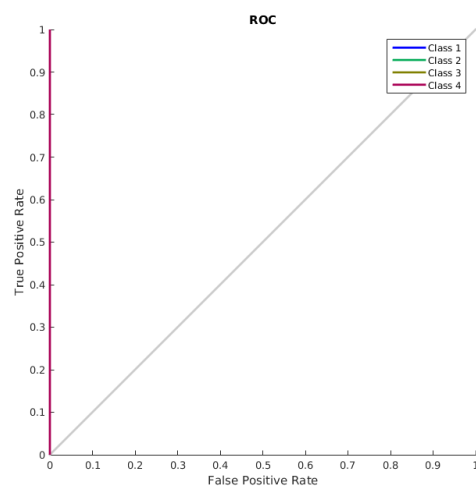
(a) Base de dados KTH



(b) Base de dados UCF101



(c) Base de dados Weizmann

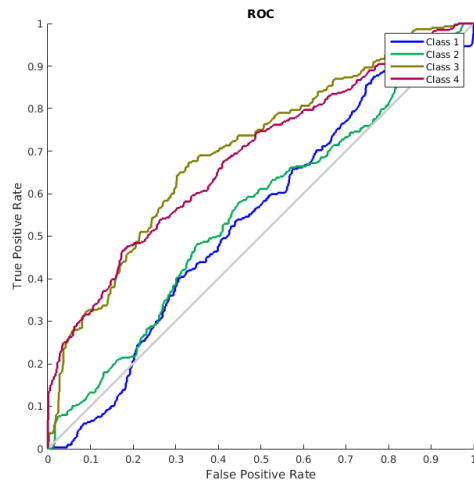


(d) Base de dados YouTube

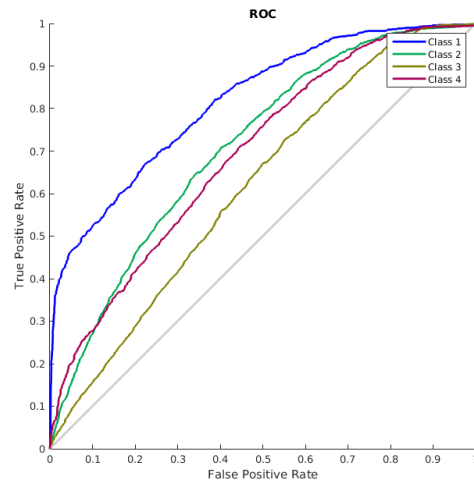
Figura 5.2: Curvas ROC do método V-STIP para o cenário de treinamento CT_1 utilizando a classificação de redes neurais de reconhecimento de padrões.

5.2.2 Redes Neurais de Ajustes de Funções

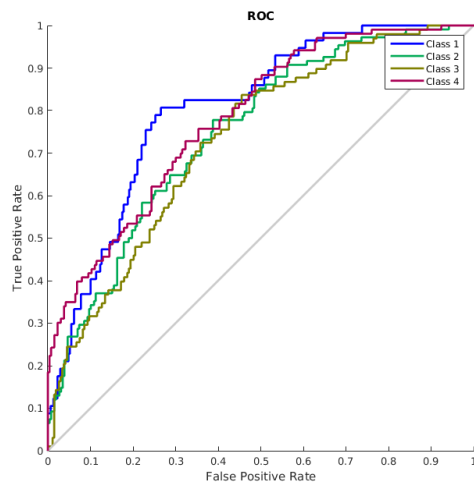
Os gráficos das curvas ROC dos melhores cenários para a rede neural de ajustes de funções estão apresentadas nas Figuras 5.3 e 5.4.



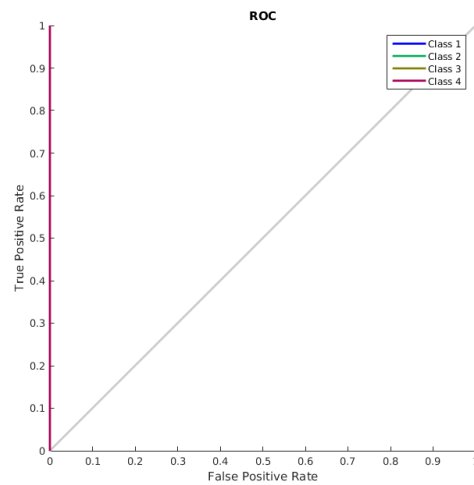
(a) Base de dados KTH



(b) Base de dados UCF101

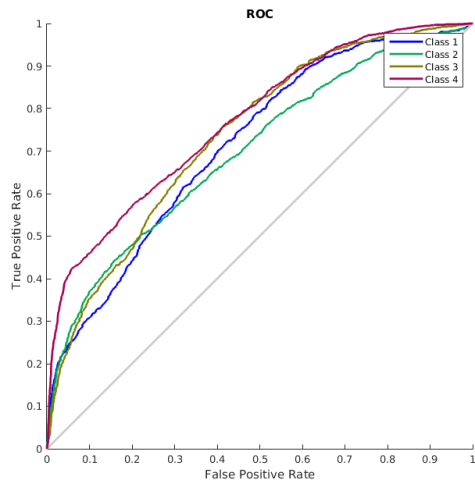


(c) Base de dados Weizmann

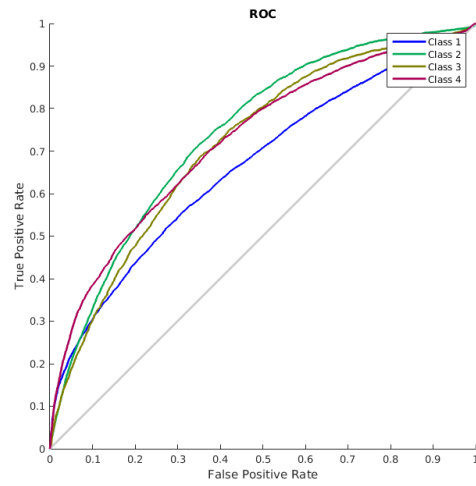


(d) Base de dados YouTube

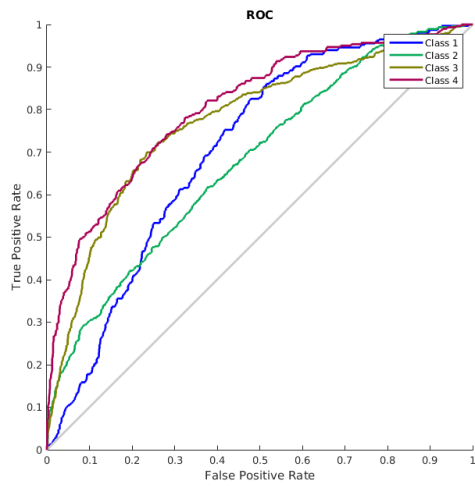
Figura 5.3: Curvas ROC do método C-STIP para o cenário de treinamento CT_6 utilizando a classificação de redes neurais de ajustes de funções.



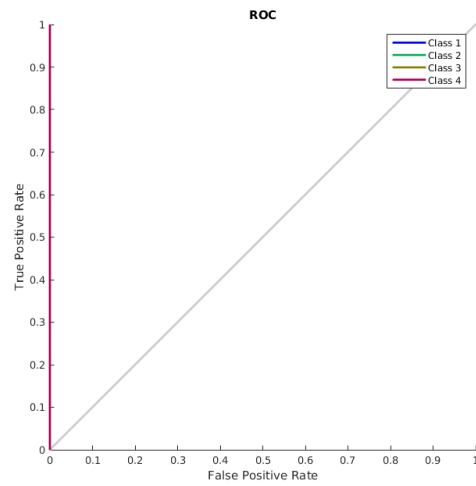
(a) Base de dados KTH



(b) Base de dados UCF101



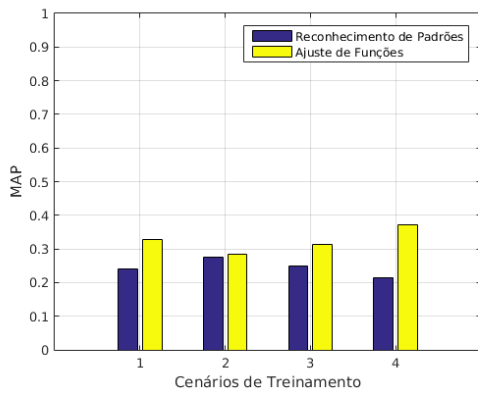
(c) Base de dados Weizmann



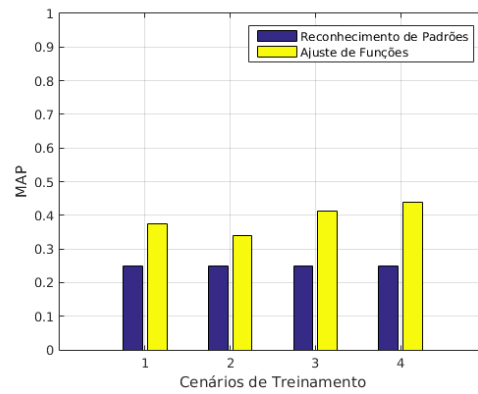
(d) Base de dados YouTube

Figura 5.4: Curvas ROC do método V-STIP para o cenário de treinamento CT_1 utilizando a classificação de redes neurais de ajustes de funções.

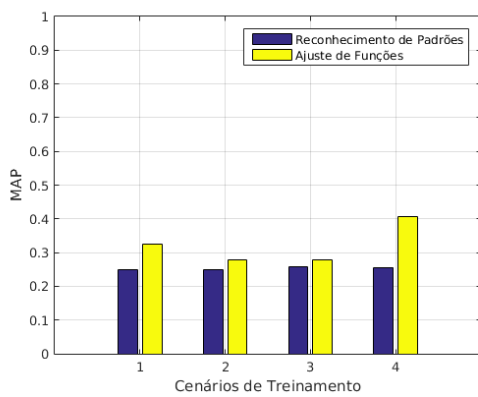
Os gráficos encontrados nas Figuras 5.5 e 5.6 apresentam comparativos da métrica MAP entre a rede com reconhecimento de padrões e a rede com ajustes de funções.



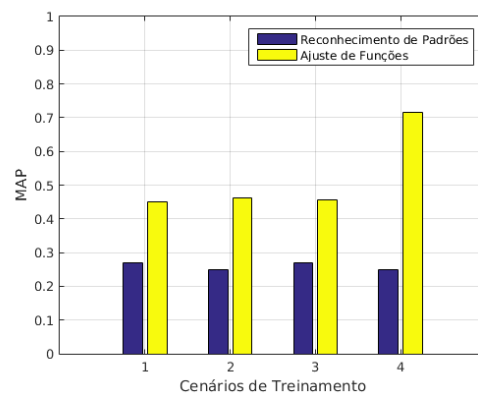
(a) Base KTH - C-STIP



(b) Base UCF101 - C-STIP

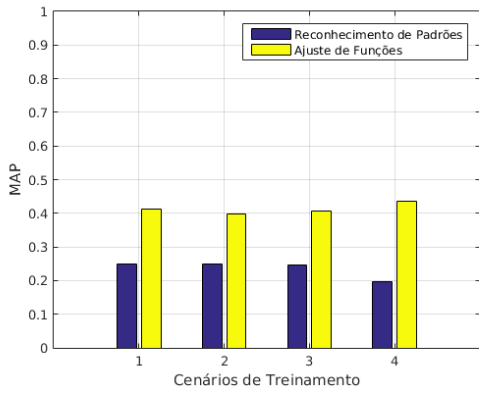


(c) Base Weizmann - C-STIP

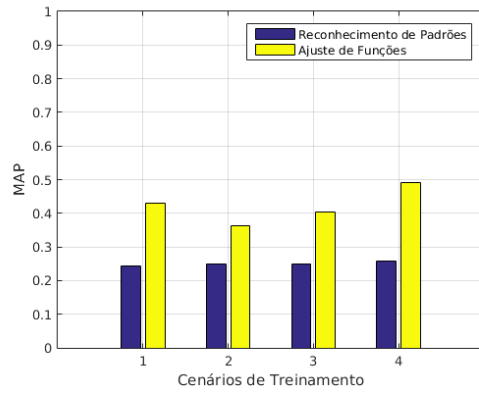


(d) Base YouTube - C-STIP

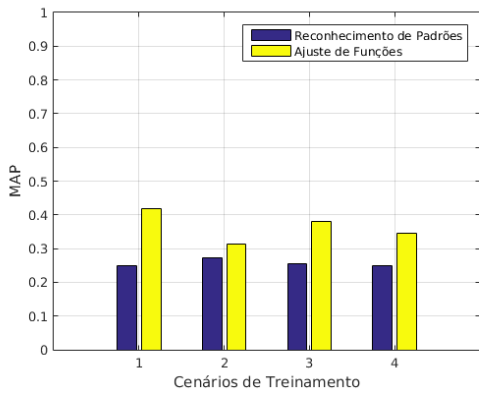
Figura 5.5: Comparação da classificação de C-STIP entre Rede Neural para Reconhecimento de Padrões e para Ajustes de Funções.



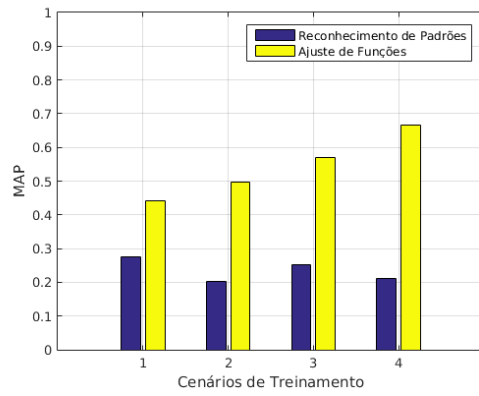
(a) Base KTH - V-STIP



(b) Base UCF101 - V-STIP



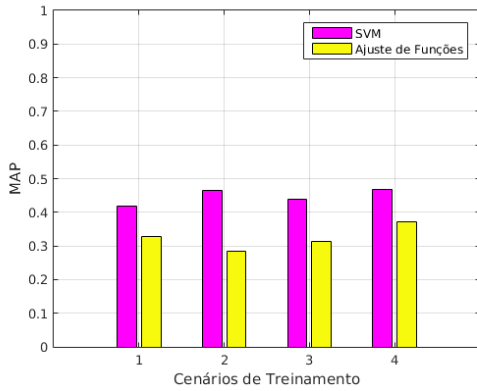
(c) Base Weizmann - V-STIP



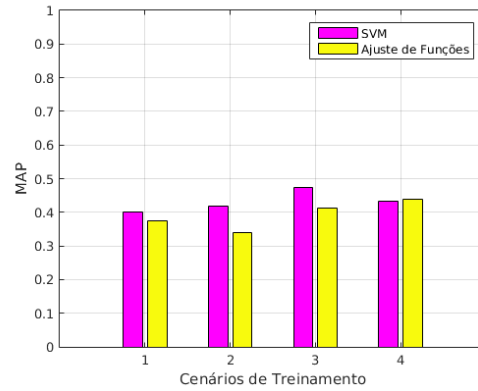
(d) Base YouTube - V-STIP

Figura 5.6: Comparação da classificação de V-STIP entre Rede Neural para Reconhecimento de Padrões e para Ajustes de Funções.

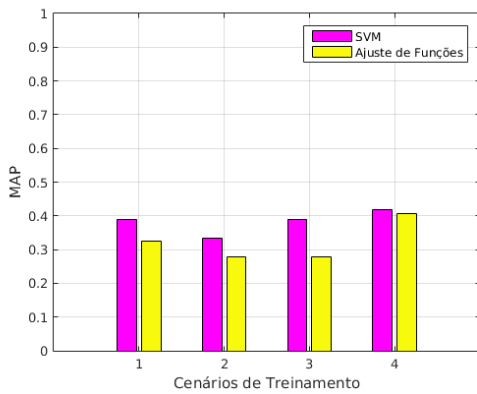
As Figuras 5.7 e 5.8 mostram a comparação entre os MAPs da melhor RNA e do SVM para C-STIP e V-STIP, respectivamente.



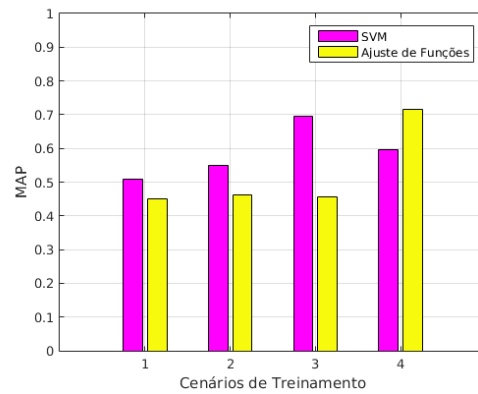
(a) Base KTH - C-STIP



(b) Base UCF101 - C-STIP

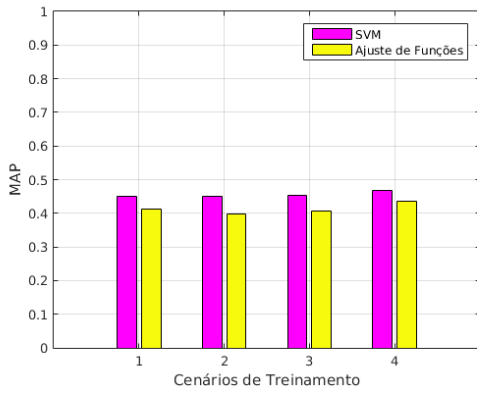


(c) Base Weizmann - C-STIP

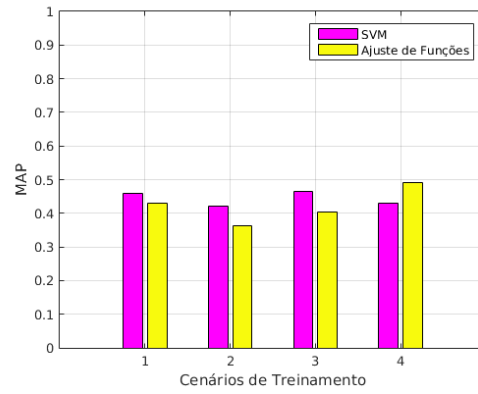


(d) Base YouTube - C-STIP

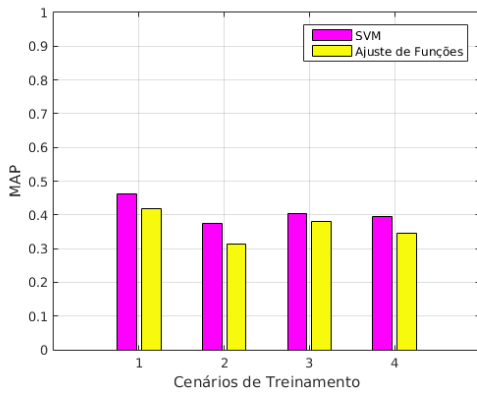
Figura 5.7: Comparação da classificação de C-STIP entre Rede Neural para Ajustes de Funções e SVM.



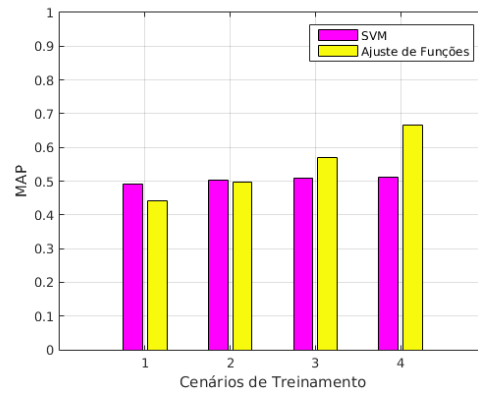
(a) Base KTH - V-STIP



(b) Base UCF101 - V-STIP



(c) Base Weizmann - V-STIP



(d) Base YouTube - V-STIP

Figura 5.8: Comparação da classificação de V-STIP entre Rede Neural para Ajustes de Funções e SVM.

5.3 Discussões dos Resultados

No primeiro cenário de treinamento, observando a Figura 3.4 e a Tabela 5.2, é possível notar que o processo de classificação da base de dados KTH, mesmo com alguns falsos positivos, foi capaz de classificar corretamente a maioria dos vídeos em cada classe. Este resultado era esperado, devido à simplicidade da base de dados, em que a ação humana é realizada em ambientes mais simples e com elevado contraste entre objeto e plano de fundo.

A correta previsão das classes ocorre até o CT_3 , onde observa-se um insistente equívoco entre as classes *boxing* e *handwaving*. Tal incorreção pode ser explicada pela posição do indivíduo nos vídeos, que não possui movimentação alguma na parte inferior do corpo e o movimento na parte superior do corpo é semelhante.

De maneira similar, a confusão entre as classes *punch* e *jumping jack* da base de dados UCF101, que pode ser observada em todos os cenários, se dá pela semelhança de movimentação tanto na parte superior quanto na inferior nos vídeos utilizados.

A base de dados Weizmann, apesar de ser considerada simples pois há a presença de câmeras estáticas e atores, tem a presença de muito ruído em seu plano de fundo, desorientando de certa forma os algoritmos propostos. No entanto, mesmo com a presença desse agravante, a maioria das classes pode ser classificada, notando apenas que as classes *bend* e *jump in place* se confundiram em praticamente todas as matrizes de confusão. Ademais, o uso do algoritmo V-STIP se mostrou mais eficiente para a classificação dessas duas classes, pois quando o movimento de se curvar (*bend*) é realizado, pontos STIPs com variações similares na parte inferior do corpo geram um número maior de saliências nessa região.

A base YouTube apresenta uma confusão entre as classes *juggling* e *shooting*. Ambas as classes apresentam vídeos com um objeto de formato e cor similar saltando sucessivamente em locais semelhantes, causando uma confusão no classificador SVM.

Analisando os resultados globais encontrados nas Tabelas 5.2 a 5.8 e o método C-STIP como referência, nota-se que para a base KTH, o melhor cenário de treinamento foi o CT_7 , já para a base UCF101 e para a base YouTube, o cenário mais adequado é o CT_5 . A base Weizmann possui resultados superiores no CT_6 .

Considerando os resultados globais para o método V-STIP, a base KTH possui um MAP mais elevado no cenário CT_7 . Para a base UCF101, tanto a configuração do CT_1 quanto do CT_5 são compatíveis para um melhor resultado. Em relação as bases Weizmann e YouTube, os melhores cenários estão em CT_1 e CT_4 , respectivamente.

Todas as bases de dados possuíam vídeos com durações distintas, como apresentado Seção 4.1, e a metodologia proposta foi capaz de classificar corretamente os vídeos, mesmo com o tempo variável, demonstrando a robustez da técnica.

Quanto às RNAs, as curvas ROC descrevem a capacidade discriminativa do conjunto de dados e quanto maior é a região existente entre a diagonal e a curva mostrada, aproximando-se do canto superior esquerdo, melhor é o resultado do teste. O treinamento pode ser realizado, cumprindo o objetivo inicial e, a partir da comparação entre as curvas ROC apresentadas neste Capítulo (Figuras 5.1 a 5.4) e no Apêndice 6, nota-se que entre os dois modelos sugeridos de redes neurais, o que mais se adequou ao problema proposto foi a rede neural de ajuste de funções.

Baseando-se nessas informações, os MAPs da rede neural de ajuste de funções e do SVM são colacionados, visando definir o classificador ideal para a metodologia proposta. Observa-se que para a maioria dos cenários testados e para ambos os métodos, o classificador mais simples, SVM, alcançou resultados de 3% a 9% superiores aos concorrentes de RNAs, sendo considerado então, o melhor classificador para a metodologia proposta.

Considerando o SVM como melhor classificador e a base KTH em seu melhor cenário de treinamento, o CT_7 , a comparação das acurácias entre o estado-da-arte, trabalhos supracitados, C-STIP e V-STIP é feita na Tabela 5.14

Tabela 5.14: Comparação entre as acurácias da base de dados KTH para os trabalhos de Laptev [12] (estado-da-arte), Oikonomopoulos [8], Dollar[7], Willems [10] e Wong [9].

[12]	[8]	[10]	[7]	[9]	C-STIP	V-STIP
29.79%	66.90%	84.26%	85.92%	86.62%	58.04%	76.85%

Ao explorar o método inicial de Laptev [12], foi possível melhorar os resultados do estado-da-arte com uma nova abordagem para a mesma técnica, aumentando a confiabilidade nos pontos de interesse e convertendo-os de detectores para descritores. A adaptação do estado-da-arte também se prova mais eficiente do que os pontos de saliência espaço-temporais, apresentado por Oikonomopoulos [8]. Os trabalhos de Willems [10] e Dollar [7] possuem descritores suplementares em sua essência, como histogramas locais, e ainda utilizam de técnicas para reduzir a dimensionalidade dos pontos obtidos, como a análise de componentes principais [62].

Ademais, a abordagem de Wong [9], além de usar todas as táticas supracitadas, remove grande parte dos ruídos em vídeo, diminuindo a região de busca dos pontos de interesse. Entretanto, as acurácias obtidas pelas metodologias propostas mostradas na Tabela 5.1 provam que a comparação entre técnicas que dispõem de excesso de estratégias para melhorarem suas performances é válida e competitiva, possuindo uma baixa diferença que varia de 8% a 10%.

Capítulo 6

Conclusão

A hipótese de que o reconhecimento de ações em vídeo pode ser realizado utilizando descritores STIP sem adicionais de outros descritores foi proposta e analisada neste trabalho. A avaliação de técnicas estado-da-arte de descritores espaço-temporais foi feita usando três classificadores distintos para reconhecimento de ações humanas em vídeo. Duas implementações destes descritores foram propostas e aplicadas em quatro bases de dados completamente diferentes para classificar ações, independente da resolução e duração de cada vídeo.

A proposta de utilização dos STIPs como descritores e não como detectores em suas duas variações (C-STIP e V-STIP) é válida, pois a etapa de classificação conseguiu classificar corretamente a maioria das classes propostas, indicando que seu uso pode ser considerado como uma proposta alternativa para o problema de reconhecimento de ações humanas em vídeos. Além disso, os resultados gerados são considerados competitivos quando comparados ao estado-da-arte e trabalhos anteriores.

Considerando os vídeos sem corte e com mais ações no vídeo, também denominados por *unclipped videos*, que são vídeos com diferentes ações ocorrendo ao mesmo tempo, é possível afirmar que a metodologia proposta consegue, do mesmo modo, classificar a ação principal da cena, uma vez que duas bases de dados testadas, UCF101 e YouTube, possuem de certa forma, essa característica, e seus melhores MAPs foram de 47% e 61% para o método C-STIP, na devida ordem, e de 46% e 52% para a técnica de V-STIP, respectivamente.

O estudo dos descritores espaço-temporais permitiu o entendimento aprofundado do desenvolvimento destas técnicas e os resultados aqui apresentados mostram que ainda há espaço para melhorias nesta área de visão computacional que não englobam o uso de classificadores complexos, como o *Deep Learning*. Como contribuição adicional, os resultados obtidos servem também como uma diretiva dos parâmetros de entrada para essas técnicas, otimizando a etapa de ajuste para trabalhos futuros.

Outrossim, novos testes serão realizados com classificadores mais robustos e complexos, como a técnica de rede neural convolucional de *Deep Learning*, que mesmo considerado um

classificador antigo, vem apresentando resultados significativos na literatura recente. Além disso, uma adaptação relacionada às informações dinâmicas das técnicas implementadas vem sendo aperfeiçoada de maneira que a captação dos pontos seja melhor efetuada e, conseqüentemente, tenha um progresso nos resultados apresentados.

Referências Bibliográficas

- [1] T. Kanade, “Region segmentation: Signal vs semantics,” *Computer Graphics and Image Processing*, vol. 13, no. 4, pp. 279 – 297, 1980.
- [2] K. Rohr, “Towards model-based recognition of human movements in image sequences,” *CVGIP: Image Underst.*, vol. 59, pp. 94–115, Jan. 1994.
- [3] D. M. Gavrila and L. S. Davis, “3-d model-based tracking of humans in action: a multi-view approach,” in *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR '96, 1996 IEEE Computer Society Conference on*, pp. 73–80, Jun 1996.
- [4] A. F. Bobick and J. W. Davis, “The recognition of human movement using temporal templates,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 257–267, Mar 2001.
- [5] S. Carlsson and J. Sullivan, “Action Recognition by Shape Matching to Key Frames,” in *Workshop on Models versus Exemplars in Computer Vision*, 2001.
- [6] A. Yilmaz and M. Shah, “Actions sketch: A novel action representation,” in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 984–989, IEEE, 2005.
- [7] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, “Behavior recognition via sparse spatio-temporal features,” in *Proceedings of the 14th International Conference on Computer Communications and Networks, ICCCN '05, (Washington, DC, USA)*, pp. 65–72, IEEE Computer Society, 2005.
- [8] A. Oikonomopoulos, I. Patras, and M. Pantic, “Spatiotemporal salient points for visual recognition of human actions,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 36, no. 3, pp. 710–719, 2005.
- [9] S.-F. Wong and R. Cipolla, “Extracting spatiotemporal interest points using global information,” in *2007 IEEE 11th International Conference on Computer Vision*, pp. 1–8, IEEE, 2007.
- [10] G. Willems, T. Tuytelaars, and L. Van Gool, “An efficient dense and scale-invariant spatio-temporal interest point detector,” in *European conference on computer vision*, pp. 650–663, Springer, 2008.

- [11] B. Chakraborty, M. B. Holte, T. B. Moeslund, and J. González, “Selective spatio-temporal interest points,” *Computer Vision and Image Understanding*, vol. 116, no. 3, pp. 396 – 410, 2012. Special issue on Semantic Understanding of Human Behaviors in Image Sequences.
- [12] I. Laptev and T. Lindeberg, “Interest point detection and scale selection in space-time,” in *International Conference on Scale-Space Theories in Computer Vision*, pp. 372–387, Springer, 2003.
- [13] I. Laptev, “On space-time interest points,” *International Journal of Computer Vision*, vol. 64, no. 2-3, pp. 107–123, 2005.
- [14] S. HAYKIN, *Redes Neurais - 2ed.* BOOKMAN COMPANHIA ED, 2001.
- [15] D. Gavrila, L. Davis, *et al.*, “Towards 3-d model-based tracking and recognition of human movement: a multi-view approach,” Citeseer.
- [16] J. Davis and A. Bobick, “The representation and recognition of action using temporal templates,” pp. 928–934, 1997.
- [17] I. Laptev and T. Lindeberg, “Space-time interest points,” in *IN ICCV*, pp. 432–439, 2003.
- [18] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, “Actions as space-time shapes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 2247–2253, Dec 2007.
- [19] J. C. Niebles, H. Wang, and L. Fei-Fei, “Unsupervised learning of human action categories using spatial-temporal words,” *International journal of computer vision*, vol. 79, no. 3, pp. 299–318, 2008.
- [20] H. Zhang and L. E. Parker, “4-dimensional local spatio-temporal features for human activity recognition,” in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2044–2049, IEEE, 2011.
- [21] A. Kovashka and K. Grauman, “Learning a hierarchy of discriminative space-time neighborhood features for human action recognition,” in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 2046–2053, IEEE, 2010.
- [22] T.-H. Yu, T.-K. Kim, and R. Cipolla, “Real-time action recognition by spatiotemporal semantic and structural forests.”
- [23] M. J. Black, “Explaining optical flow events with parameterized spatio-temporal models,” in *IEEE Proc. Computer Vision and Pattern Recognition, CVPR’99*, (Fort Collins, CO), pp. 326–332, IEEE, 1999.

- [24] W. Iwun Lu and J. J. Little, "Tracking and recognizing actions at a distance," in *in: ECCV Workshop on Computer Vision Based Analysis in Sport Environments*, pp. 49–60.
- [25] R. Goldenberg, R. Kimmel, E. Rivlin, and M. Rudzsky, "Behavior classification by eigendecomposition of periodic motions," *Pattern Recogn.*, vol. 38, pp. 1033–1043, July 2005.
- [26] A. Dehghan, O. Oreifej, and M. Shah, "Complex event recognition using constrained low-rank representation," *Image and Vision Computing*, vol. 42, pp. 13–21, 2015.
- [27] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 886–893, IEEE, 2005.
- [28] S. M. Smith, "Asset-2: Real-time motion segmentation and shape tracking," in *Computer Vision, 1995. Proceedings., Fifth International Conference on*, pp. 237–244, IEEE, 1995.
- [29] I. Laptev and T. Lindeberg, "Velocity adaptation of space-time interest points," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 1, pp. 52–56, IEEE, 2004.
- [30] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local svm approach," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 3, pp. 32–36, IEEE, 2004.
- [31] D. D. Dawn and S. H. Shaikh, "A comprehensive survey of human action recognition with spatio-temporal interest point (stip) detector," *The Visual Computer*, vol. 32, no. 3, pp. 289–306, 2016.
- [32] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," in *The Tenth IEEE International Conference on Computer Vision (ICCV'05)*, pp. 1395–1402, 2005.
- [33] I. Laptev, "Local spatio-temporal image features for motion interpretation," 2004.
- [34] M. B. Clowes, "On seeing things," *Artificial intelligence*, vol. 2, no. 1, pp. 79–116, 1971.
- [35] T. Kanade, "Model representations and control structures in image understanding.,"
- [36] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.
- [37] J. M. Rehg and T. Kanade, "Model-based tracking of self-occluding articulated objects," in *Computer Vision, 1995. Proceedings., Fifth International Conference on*, pp. 612–617, IEEE, 1995.

- [38] C. Canton-Ferrer, J. R. Casas, and M. Pardàs, “Human model and motion based 3d action recognition in multiple view scenarios,” in *Signal Processing Conference, 2006 14th European*, pp. 1–5, Sept 2006.
- [39] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, “Performance of optical flow techniques,” *Int. J. Comput. Vision*, vol. 12, pp. 43–77, Feb. 1994.
- [40] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.
- [41] L. Zelnik-Manor and M. Irani, “Event-based analysis of video,” in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 2, pp. II–123, IEEE, 2001.
- [42] K. Mikolajczyk and C. Schmid, “Scale & affine invariant interest point detectors,” *International Journal of Computer Vision*, vol. 60, no. 1, pp. 63–86, 2004.
- [43] J. Liu, S. Ali, and M. Shah, “Recognizing human actions using multiple features,” 2008.
- [44] C. Harris and M. Stephens, “A combined corner and edge detector.,” in *Alvey vision conference*, vol. 15, p. 50, Citeseer, 1988.
- [45] R. Laganière, *OpenCV 2 Computer Vision Application Programming Cookbook: Over 50 recipes to master this library of programming functions for real-time computer vision*. Packt Publishing Ltd, 2011.
- [46] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Prentice Hall, 2 ed., 2002.
- [47] E. Trucco and A. Verri, *Introductory techniques for 3-D computer vision*, vol. 201. 1998.
- [48] T. Lindeberg, “Time-recursive velocity-adapted spatio-temporal scale-space filters,” in *European Conference on Computer Vision*, pp. 52–67, Springer, 2002.
- [49] B. D. Lucas, T. Kanade, *et al.*, “An iterative image registration technique with an application to stereo vision.,” 1981.
- [50] K. Soomro, A. R. Zamir, and M. Shah, “UCF101: A dataset of 101 human actions classes from videos in the wild,” 2012.
- [51] J. Liu, J. Luo, and M. Shah, “Recognizing realistic actions from videos “in the wild”,” in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 1996–2003, IEEE, 2009.
- [52] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson, *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd ed., 2001.

- [53] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [54] A. Ben-Hur, D. Horn, H. T. Siegelmann, and V. Vapnik, "Support vector clustering," *Journal of machine learning research*, vol. 2, no. Dec, pp. 125–137, 2001.
- [55] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.
- [56] MATLAB, *version 7.10.0 (R2010a)*. Natick, Massachusetts: The MathWorks Inc., 2010.
- [57] M. F. Møller, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural networks*, vol. 6, no. 4, pp. 525–533, 1993.
- [58] D. M. Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation," 2011.
- [59] C. J. V. Rijsbergen, *Information Retrieval*. Newton, MA, USA: Butterworth-Heinemann, 2nd ed., 1979.
- [60] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [61] S. de Bioestatística e Informática Médica, "Avaliação de testes diagnósticos,"
- [62] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philosophical Magazine*, vol. 2, pp. 559–572, 1901.
- [63] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain," *Psychological review*, vol. 65, no. 6, p. 386, 1958.
- [64] D. E. Rumelhart *et al.*, *Parallel distributed processing*, vol. 1.

APÊNDICES

Neste apêndice são apresentados as matrizes de confusão referentes às classificações SVM, bem como os gráficos ROC das redes neurais de reconhecimento de padrões e de ajustes de funções.

6.1 Matrizes de Confusão

As matrizes de confusão a seguir foram usadas para a discussão dos resultados no Capítulo 5.

	Boxing	Handwaving	Running	Walking
Boxing	42%	34%	8%	16%
Handwaving	30%	39%	16%	15%
Running	19%	22%	47%	12%
Walking	24%	26%	10%	40%

	Boxing	Handwaving	Running	Walking
Boxing	50%	34.5%	5%	10.5%
Handwaving	34%	42%	13%	11%
Running	20%	22%	51%	7%
Walking	28%	28%	7%	37%

(a) Precisões de C-STIP

(b) Precisões de V-STIP

Figura 6.1: Matriz de confusão do CT_1 para a base de dados KTH.

	Boxing	Handwaving	Running	Walking
Boxing	43%	35%	8.5%	13.5%
Handwaving	26.5%	34%	27.5%	12%
Running	16%	19%	56%	9%
Walking	24%	27.5%	7.5%	41%

(a) Precisões de C-STIP

	Boxing	Handwaving	Running	Walking
Boxing	52%	36%	4%	8%
Handwaving	29%	42%	22%	7%
Running	21.5%	22.5%	51%	5%
Walking	30%	31%	4%	35%

(b) Precisões de V-STIP

Figura 6.2: Matriz de confusão do CT_2 para a base de dados KTH.

	Boxing	Handwaving	Running	Walking
Boxing	42%	35%	8%	15%
Handwaving	30%	40%	15%	15%
Running	19%	23%	47%	11%
Walking	23%	26%	11%	40%

(a) Precisões de C-STIP

	Boxing	Handwaving	Running	Walking
Boxing	88%	4%	2%	6%
Handwaving	37%	45.5%	8.5%	9%
Running	23%	23%	48%	6%
Walking	30%	28.5%	6%	35.5%

(b) Precisões de V-STIP

Figura 6.3: Matriz de confusão do CT_3 para a base de dados KTH.

	Boxing	Handwaving	Running	Walking
Boxing	48%	26%	8%	18%
Handwaving	38%	27%	13%	22%
Running	19%	17.5%	51%	12.5%
Walking	22%	18%	11%	49%

(a) Precisões de C-STIP

	Boxing	Handwaving	Running	Walking
Boxing	58%	27%	5.5%	9.5%
Handwaving	46%	34%	9%	11%
Running	20%	19%	55%	6%
Walking	30%	23.5%	8%	38.5%

(b) Precisões de V-STIP

Figura 6.4: Matriz de confusão do CT_4 para a base de dados KTH.

	Boxing	Handwaving	Running	Walking
Boxing	48%	38%	5%	9%
Handwaving	32%	46%	14%	8%
Running	18%	23%	52%	7%
Walking	26%	31%	12%	30%

(a) Precisões de C-STIP

	Boxing	Handwaving	Running	Walking
Boxing	52%	38%	3%	6%
Handwaving	37%	43%	14%	5%
Running	22%	23%	51%	4%
Walking	31%	28%	8%	34%

(b) Precisões de V-STIP

Figura 6.5: Matriz de confusão do CT_5 para a base de dados KTH.

	Boxing	Handwaving	Running	Walking
Boxing	50%	37%	5%	8%
Handwaving	41%	47%	4%	8%
Running	15%	20%	61%	4%
Walking	29%	31%	10%	30%

(a) Precisões de C-STIP

	Boxing	Handwaving	Running	Walking
Boxing	62%	27%	4%	7%
Handwaving	48%	44%	3%	5%
Running	29%	20%	46%	5%
Walking	35%	22%	7%	37%

(b) Precisões de V-STIP

Figura 6.6: Matriz de confusão do CT_6 para a base de dados KTH.

	Boxing	Handwaving	Running	Walking
Boxing	52%	35%	6%	7%
Handwaving	43%	47%	5%	6%
Running	12%	17%	68%	4%
Walking	29%	32%	12%	27%

(a) Precisões de C-STIP

	Boxing	Handwaving	Running	Walking
Boxing	76%	15%	5%	4%
Handwaving	53%	42%	3%	2%
Running	2%	4%	93%	1%
Walking	28%	25%	9%	38%

(b) Precisões de V-STIP

Figura 6.7: Matriz de confusão do CT_7 para a base de dados KTH.

	Biking	Jumping Jack	Punch	Walking with Dog
Biking	50%	16%	21%	13%
Jumping Jack	15%	33%	35%	17%
Punch	11%	30%	43%	16%
Walking with Dog	21%	16%	29%	34%

(a) Precisões de C-STIP

	Biking	Jumping Jack	Punch	Walking with Dog
Biking	45%	31%	12%	12%
Jumping Jack	26%	47%	14%	13%
Punch	30%	9%	44%	17%
Walking with Dog	25%	7%	20%	48%

(b) Precisões de V-STIP

Figura 6.8: Matriz de confusão do CT_1 para a base de dados UCF101.

	Biking	Jumping Jack	Punch	Walking with Dog
Biking	52%	14%	18%	16%
Jumping Jack	14%	39%	33%	14%
Punch	10%	37%	39%	14%
Walking with Dog	14%	17%	32%	37%

(a) Precisões de C-STIP

	Biking	Jumping Jack	Punch	Walking with Dog
Biking	49%	12%	19%	20%
Jumping Jack	15%	33%	37%	15%
Punch	11%	29%	44%	16%
Walking with Dog	13.5%	14%	30.5%	42%

(b) Precisões de V-STIP

Figura 6.9: Matriz de confusão do CT_2 para a base de dados UCF101.

	Biking	Jumping Jack	Punch	Walking with Dog
Biking	47%	18.5%	22%	12.5%
Jumping Jack	37%	45.5%	8.5%	9%
Punch	23%	23%	48%	6%
Walking with Dog	30%	28%	6%	36%

(a) Precisões de C-STIP

	Biking	Jumping Jack	Punch	Walking with Dog
Biking	43%	16%	26%	15%
Jumping Jack	11%	29.5%	43%	16.5%
Punch	7%	25%	45.5%	22.5%
Walking with Dog	15%	16%	29%	40%

(b) Precisões de V-STIP

Figura 6.10: Matriz de confusão do CT_3 para a base de dados UCF101.

	Biking	Jumping Jack	Punch	Walking with Dog
Biking	54%	11%	18.5%	16.5%
Jumping Jack	18%	29%	38%	15%
Punch	10%	26%	42%	22%
Walking with Dog	13%	17%	28%	42%

(a) Precisões de C-STIP

	Biking	Jumping Jack	Punch	Walking with Dog
Biking	49%	13%	20%	18%
Jumping Jack	12%	31%	43%	14%
Punch	11%	25%	42%	22%
Walking with Dog	13%	17%	28%	42%

(b) Precisões de V-STIP

Figura 6.11: Matriz de confusão do CT_4 para a base de dados UCF101.

	Biking	Jumping Jack	Punch	Walking with Dog
Biking	65%	9%	15%	11%
Jumping Jack	10%	34%	43%	14%
Punch	6%	31%	49%	14%
Walking with Dog	12%	17%	29%	43%

(a) Precisões de C-STIP

	Biking	Jumping Jack	Punch	Walking with Dog
Biking	57%	10%	20%	13%
Jumping Jack	16%	27%	42%	15%
Punch	8%	26%	53%	14%
Walking with Dog	10%	14%	27%	49%

(b) Precisões de V-STIP

Figura 6.12: Matriz de confusão do CT_5 para a base de dados UCF101.

	Biking	Jumping Jack	Punch	Walking with Dog
Biking	62%	11%	16%	11%
Jumping Jack	15%	30%	41%	13%
Punch	13%	27%	47%	13%
Walking with Dog	17%	17%	31%	35%

(a) Precisões de C-STIP

	Biking	Jumping Jack	Punch	Walking with Dog
Biking	53%	14%	21%	12%
Jumping Jack	19%	29%	38%	14%
Punch	13%	22%	51%	14%
Walking with Dog	14%	15%	32%	40%

(b) Precisões de V-STIP

Figura 6.13: Matriz de confusão do CT_6 para a base de dados UCF101.

	Biking	Jumping Jack	Punch	Walking with Dog
Biking	52%	11%	21%	16%
Jumping Jack	15%	29%	43%	14%
Punch	23%	21%	39%	18%
Walking with Dog	13%	14%	27%	46%

(a) Precisões de C-STIP

	Biking	Jumping Jack	Punch	Walking with Dog
Biking	26%	21%	33%	19%
Jumping Jack	15%	30%	43%	12%
Punch	22%	22%	39%	17%
Walking with Dog	5%	10%	19%	65%

(b) Precisões de V-STIP

Figura 6.14: Matriz de confusão do CT_7 para a base de dados UCF101.

	Bend	Jump in Place	Gallop Sideways	Skip
Bend	35%	35.5%	17.5%	12%
Jump in Place	20%	43%	20%	17%
Gallop Sideways	12.5%	22.5%	37%	28%
Skip	12%	18%	29%	41%

(a) Precisões de C-STIP

	Bend	Jump in Place	Gallop Sideways	Skip
Bend	52%	30%	11%	7%
Jump in Place	29%	45.5%	20%	5.5%
Gallop Sideways	16%	27%	53%	4%
Skip	36%	21%	9%	34%

(b) Precisões de V-STIP

Figura 6.15: Matriz de confusão do CT_1 para a base de dados Weizmann.

	Bend	Jump in Place	Gallop Sideways	Skip
Bend	23%	37%	24%	16%
Jump in Place	14%	34%	29%	23%
Gallop Sideways	10%	22%	38%	30%
Skip	9%	19%	33%	39%

(a) Precisões de C-STIP

	Bend	Jump in Place	Gallop Sideways	Skip
Bend	42%	34%	13.5%	10.5%
Jump in Place	26%	34%	21%	19%
Gallop Sideways	14.5%	20%	35.5%	30%
Skip	16%	17%	29%	38%

(b) Precisões de V-STIP

Figura 6.16: Matriz de confusão do CT_2 para a base de dados Weizmann.

	Bend	Jump in Place	Gallop Sideways	Skip
Bend	23%	39%	22%	16%
Jump in Place	17%	39%	25%	19%
Gallop Sideways	10%	26%	35%	29%
Skip	9%	22%	33%	36%

(a) Precisões de C-STIP

	Bend	Jump in Place	Gallop Sideways	Skip
Bend	41%	38%	12%	9%
Jump in Place	30%	42%	15%	13%
Gallop Sideways	16%	25%	33%	26%
Skip	16%	22%	29%	33%

(b) Precisões de V-STIP

Figura 6.17: Matriz de confusão do CT_3 para a base de dados Weizmann.

	Bend	Jump in Place	Gallop Sideways	Skip
Bend	26%	38%	20%	16%
Jump in Place	18%	39%	24%	19%
Gallop Sideways	10%	25%	35%	30%
Skip	9%	21%	32%	38%

(a) Precisões de C-STIP

	Bend	Jump in Place	Gallop Sideways	Skip
Bend	42%	38%	11%	9%
Jump in Place	30%	40.5%	16%	13.5%
Gallop Sideways	14%	23%	32%	31%
Skip	12%	19.5%	30.5%	38%

(b) Precisões de V-STIP

Figura 6.18: Matriz de confusão do CT_4 para a base de dados Weizmann.

	Bend	Jump in Place	Gallop Sideways	Skip
Bend	35%	34%	17%	14%
Jump in Place	24%	49%	16%	11%
Gallop Sideways	14%	26%	34%	26%
Skip	10%	20%	31%	39%

(a) Precisões de C-STIP

	Bend	Jump in Place	Gallop Sideways	Skip
Bend	62%	22%	8%	9%
Jump in Place	44%	35%	12%	9%
Gallop Sideways	24%	23%	28%	24%
Skip	23%	16%	25%	37%

(b) Precisões de V-STIP

Figura 6.19: Matriz de confusão do CT_5 para a base de dados Weizmann.

	Bend	Jump in Place	Gallop Sideways	Skip
Bend	48%	35%	10%	7%
Jump in Place	27%	40%	18%	14%
Gallop Sideways	15%	20%	35%	31%
Skip	12%	16%	28%	44%

(a) Precisões de C-STIP

	Bend	Jump in Place	Gallop Sideways	Skip
Bend	57%	31%	6%	6%
Jump in Place	42%	33%	13%	12%
Gallop Sideways	28%	18%	29%	25%
Skip	14%	15%	32%	39%

(b) Precisões de V-STIP

Figura 6.20: Matriz de confusão do CT_6 para a base de dados Weizmann.

	Bend	Jump in Place	Gallop Sideways	Skip
Bend	46%	24%	19%	11%
Jump in Place	26%	40%	20%	15%
Gallop Sideways	17%	20%	34%	29%
Skip	14%	14%	31%	41%

(a) Precisões de C-STIP

	Bend	Jump in Place	Gallop Sideways	Skip
Bend	21%	51%	15%	14%
Jump in Place	12%	43%	25%	20%
Gallop Sideways	9%	29%	34%	27%
Skip	5%	18%	31%	45%

(b) Precisões de V-STIP

Figura 6.21: Matriz de confusão do CT_7 para a base de dados Weizmann.

	Shooting	Diving	Juggling	Spiking
Shooting	23%	31%	43%	3%
Diving	5%	51%	41%	3%
Juggling	6%	36%	54%	4%
Spiking	1%	7%	16%	76%

(a) Precisões de C-STIP

	Shooting	Diving	Juggling	Spiking
Shooting	14%	36%	47%	3%
Diving	6.5%	49%	42.5%	2%
Juggling	7%	38%	52%	3%
Spiking	1%	4%	14%	81%

(b) Precisões de V-STIP

Figura 6.22: Matriz de confusão do CT_1 para a base de dados YouTube.

	Shooting	Diving	Juggling	Spiking
Shooting	44%	22%	32%	2%
Diving	8%	49%	41%	2%
Juggling	9%	36%	52%	3%
Spiking	2%	5%	18%	75%

(a) Precisões de C-STIP

	Shooting	Diving	Juggling	Spiking
Shooting	22%	34%	42%	2%
Diving	9%	52%	37%	2%
Juggling	12%	38%	48%	2%
Spiking	2%	3%	16%	79%

(b) Precisões de V-STIP

Figura 6.23: Matriz de confusão do CT_2 para a base de dados YouTube.

	Shooting	Diving	Juggling	Spiking
Shooting	46%	21%	31%	2%
Diving	8%	51%	39%	2%
Juggling	6%	21%	31%	42%
Spiking	1.5%	5.5%	8%	85%

(a) Precisões de C-STIP

	Shooting	Diving	Juggling	Spiking
Shooting	23%	32%	43%	2%
Diving	8.5%	53%	38%	0.5%
Juggling	6.5%	22.5%	31%	40%
Spiking	2%	5%	7%	86%

(b) Precisões de V-STIP

Figura 6.24: Matriz de confusão do CT_3 para a base de dados YouTube.

	Shooting	Diving	Juggling	Spiking
Shooting	46.5%	20.5%	31%	2%
Diving	7%	53%	37%	3%
Juggling	10%	35%	51%	4%
Spiking	2%	6%	14%	78%

(a) Precisões de C-STIP

	Shooting	Diving	Juggling	Spiking
Shooting	21.5%	31%	45%	2.5%
Diving	8%	54%	37%	1%
Juggling	11%	35%	52%	2%
Spiking	1%	4.5%	13%	81.5%

(b) Precisões de V-STIP

Figura 6.25: Matriz de confusão do CT_4 para a base de dados YouTube.

	Shooting	Diving	Juggling	Spiking
Shooting	63%	13%	23%	1%
Diving	8%	51%	39%	2%
Juggling	11%	35%	52%	2%
Spiking	1%	5%	13%	81%

(a) Precisões de C-STIP

	Shooting	Diving	Juggling	Spiking
Shooting	20%	33%	45%	2%
Diving	9%	53%	36%	2%
Juggling	13%	37%	49%	2%
Spiking	2%	4%	12%	82%

(b) Precisões de V-STIP

Figura 6.26: Matriz de confusão do CT_5 para a base de dados YouTube.

	Shooting	Diving	Juggling	Spiking
Shooting	60%	16%	23%	1%
Diving	11%	47%	41%	1%
Juggling	14%	34%	50%	2%
Spiking	3%	6%	9%	82%

(a) Precisões de C-STIP

	Shooting	Diving	Juggling	Spiking
Shooting	29%	31%	39%	1%
Diving	14%	46%	39%	1%
Juggling	16%	35%	46%	2%
Spiking	3%	6%	8%	83%

(b) Precisões de V-STIP

Figura 6.27: Matriz de confusão do CT_6 para a base de dados YouTube.

	Shooting	Diving	Juggling	Spiking
Shooting	25%	30%	43%	2%
Diving	9%	47%	42%	2%
Juggling	11%	34%	52%	3%
Spiking	3%	10%	14%	73%

(a) Precisões de C-STIP

	Shooting	Diving	Juggling	Spiking
Shooting	24%	32%	41%	3%
Diving	13%	46%	39%	3%
Juggling	15%	35%	45%	5%
Spiking	2%	4%	5%	90%

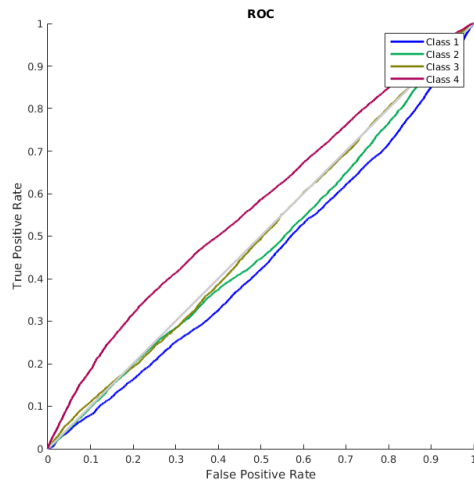
(b) Precisões de V-STIP

Figura 6.28: Matriz de confusão do CT_7 para a base de dados YouTube.

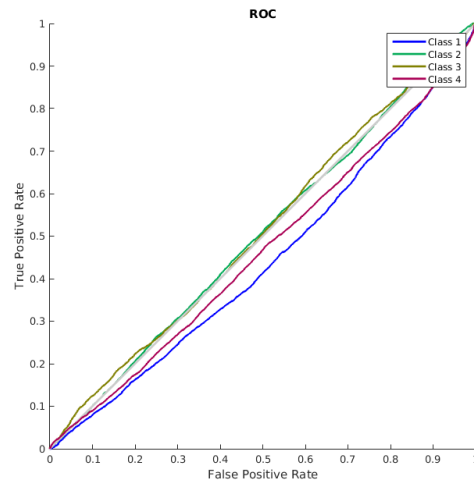
6.2 Curvas ROC

As curvas ROC restantes, isto é, as que não foram mostradas no Capítulo 5, se encontram a seguir, devidamente separadas por tipo de rede neural.

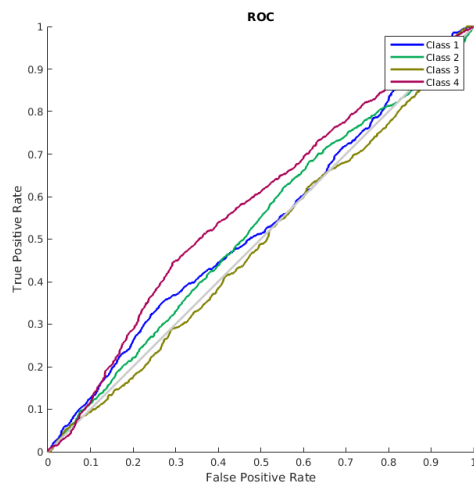
6.2.1 Curvas ROC da Rede Neural de Reconhecimento de Padrões



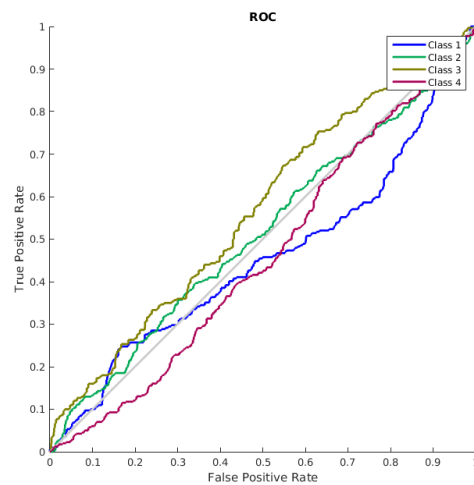
(a) Cenário de treinamento CT_1



(b) Cenário de treinamento CT_2

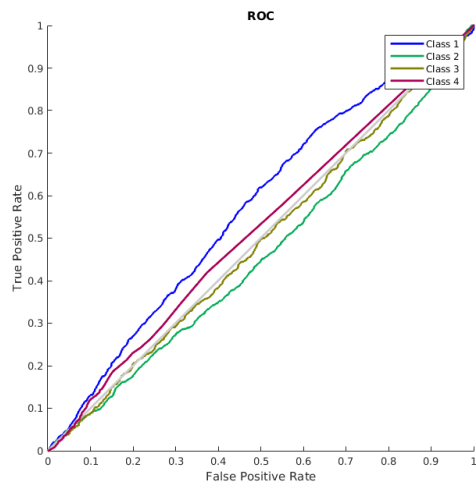


(c) Cenário de treinamento CT_5

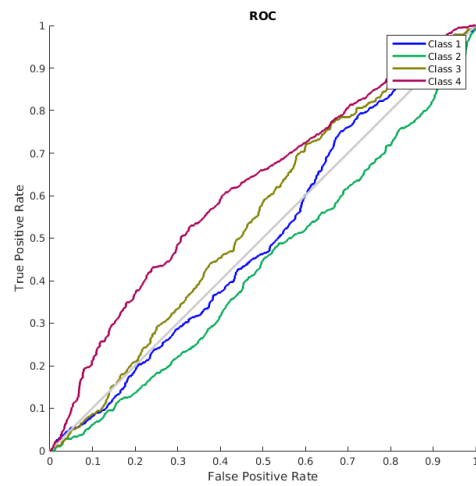


(d) Cenário de treinamento CT_6

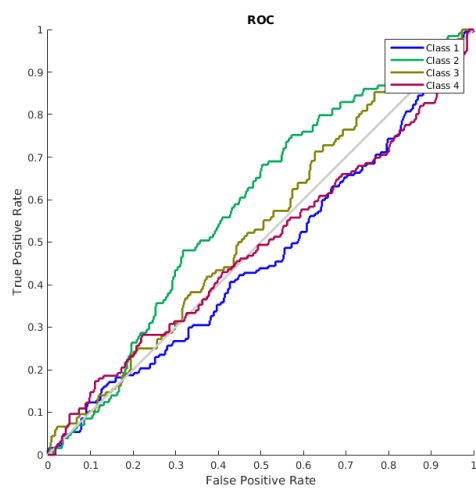
Figura 6.29: Curvas ROC do método C-STIP da base de dados KTH utilizando a classificação de redes neurais de reconhecimento de padrões.



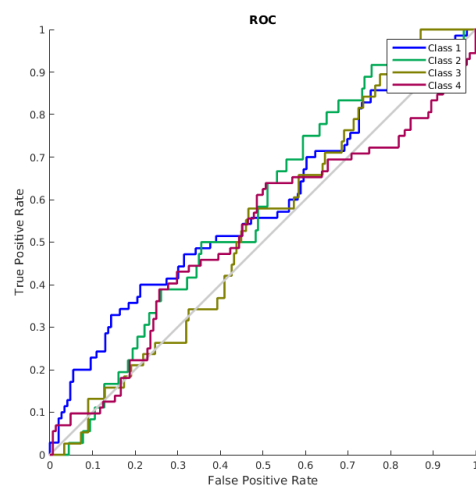
(a) Cenário de treinamento CT_1



(b) Cenário de treinamento CT_2

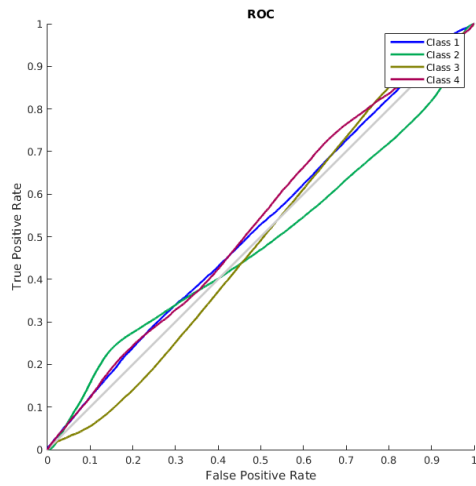


(c) Cenário de treinamento CT_5

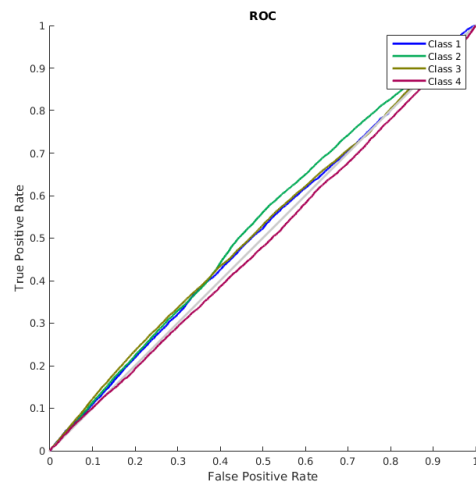


(d) Cenário de treinamento CT_6

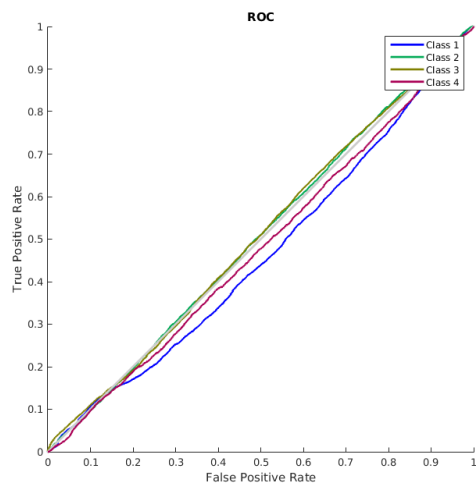
Figura 6.30: Curvas ROC do método V-STIP da base de dados KTH utilizando a classificação de redes neurais de reconhecimento de padrões.



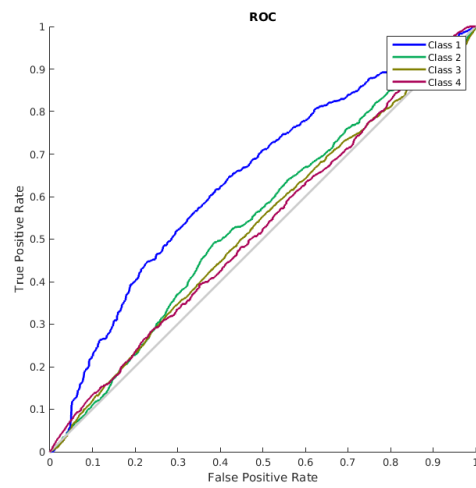
(a) Cenário de treinamento CT_1



(b) Cenário de treinamento CT_2

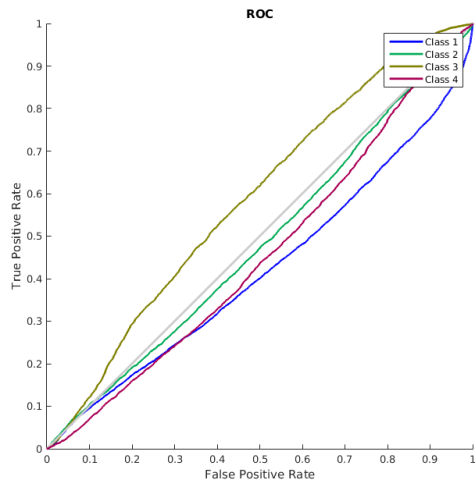


(c) Cenário de treinamento CT_5

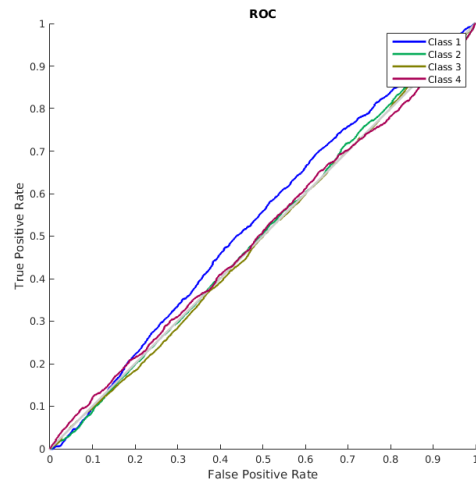


(d) Cenário de treinamento CT_6

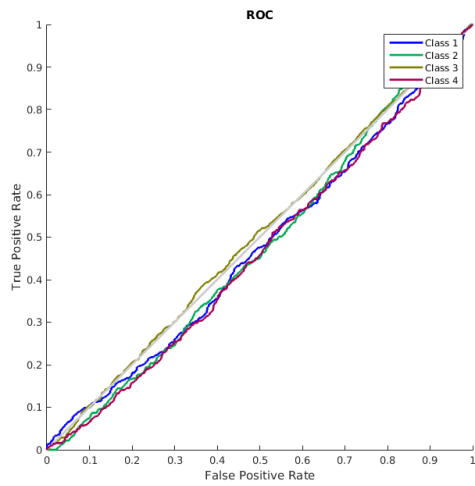
Figura 6.31: Curvas ROC do método C-STIP da base de dados UCF101 utilizando a classificação de redes neurais de reconhecimento de padrões.



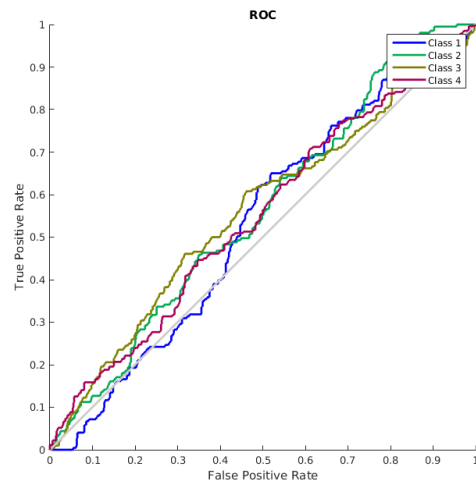
(a) Cenário de treinamento CT_1



(b) Cenário de treinamento CT_2

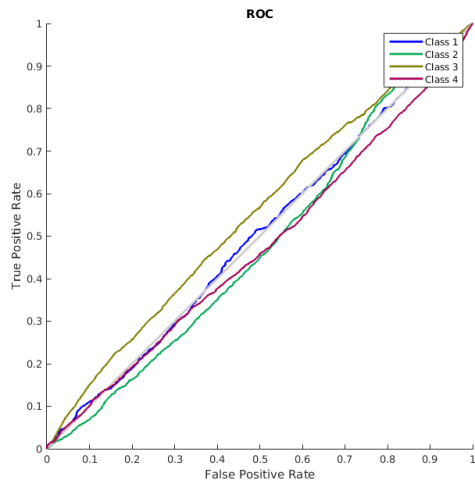


(c) Cenário de treinamento CT_5

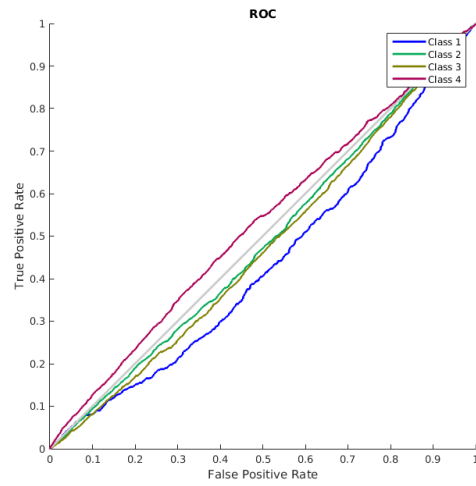


(d) Cenário de treinamento CT_6

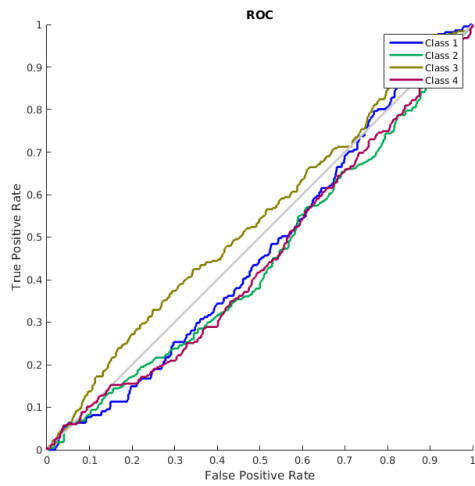
Figura 6.32: Curvas ROC do método V-STIP da base de dados UCF101 utilizando a classificação de redes neurais de reconhecimento de padrões.



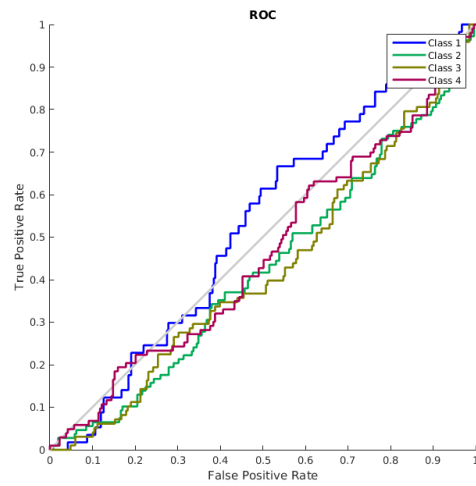
(a) Cenário de treinamento CT_1



(b) Cenário de treinamento CT_2

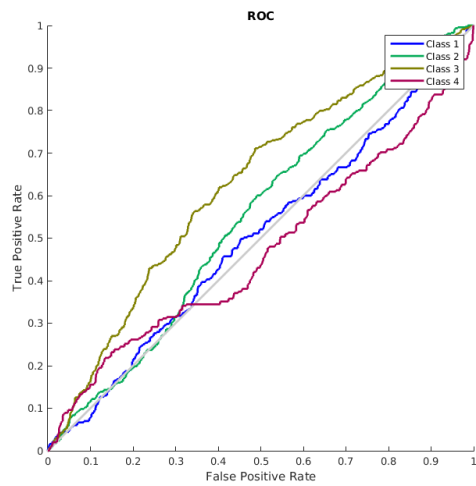


(c) Cenário de treinamento CT_5

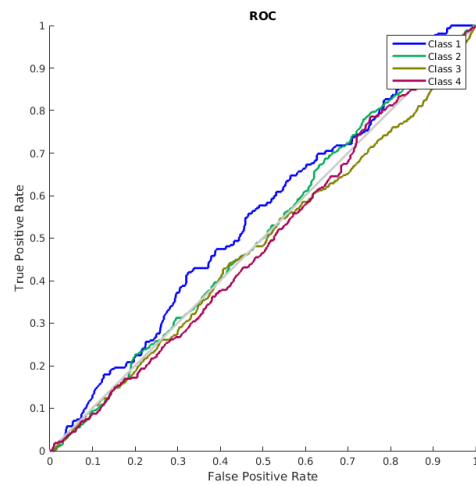


(d) Cenário de treinamento CT_6

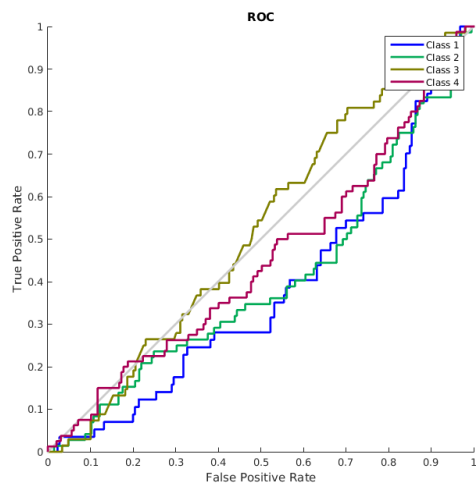
Figura 6.33: Curvas ROC do método C-STIP da base de dados Weizmann utilizando a classificação de redes neurais de reconhecimento de padrões.



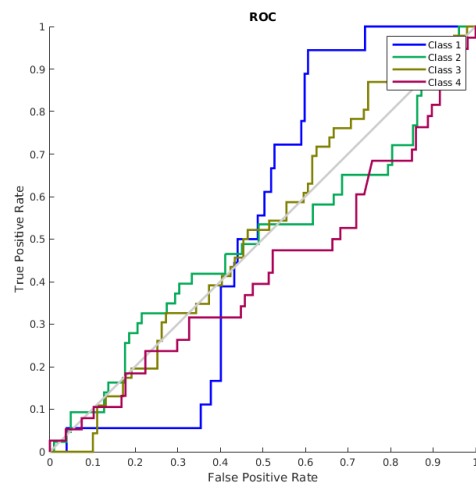
(a) Cenário de treinamento CT_1



(b) Cenário de treinamento CT_2

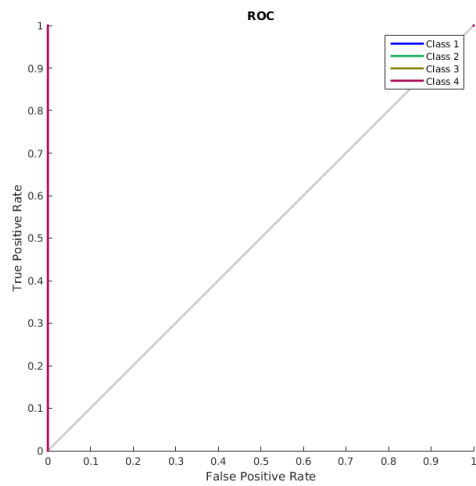


(c) Cenário de treinamento CT_5

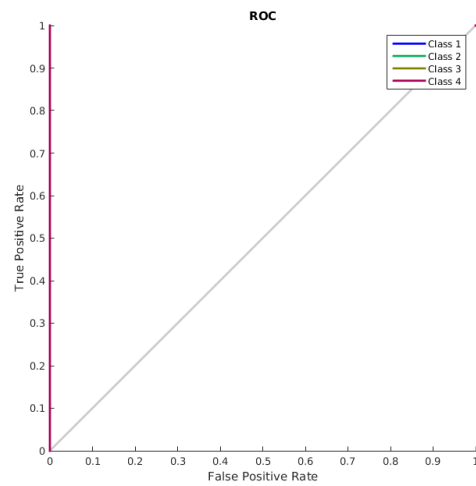


(d) Cenário de treinamento CT_6

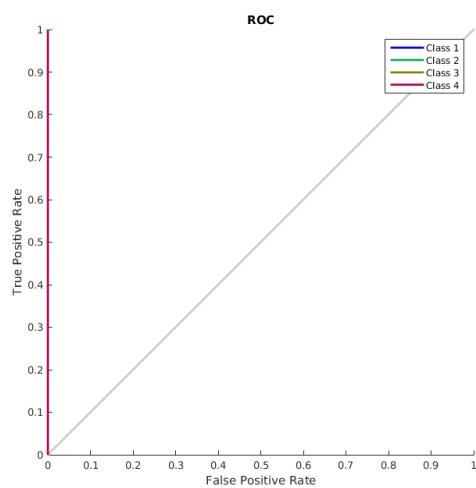
Figura 6.34: Curvas ROC do método V-STIP da base de dados Weizmann utilizando a classificação de redes neurais de reconhecimento de padrões.



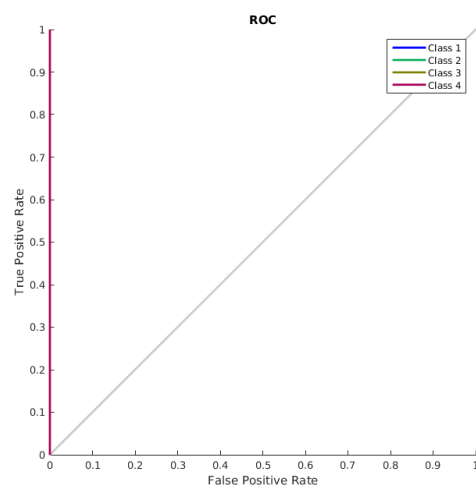
(a) Cenário de treinamento CT_1



(b) Cenário de treinamento CT_2

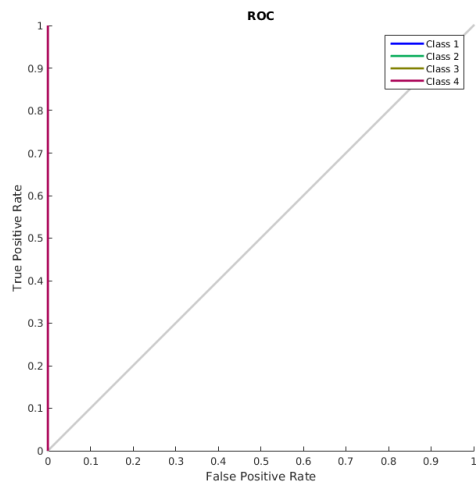


(c) Cenário de treinamento CT_5

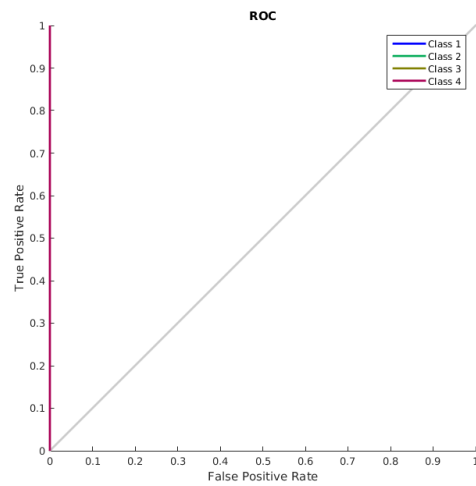


(d) Cenário de treinamento CT_6

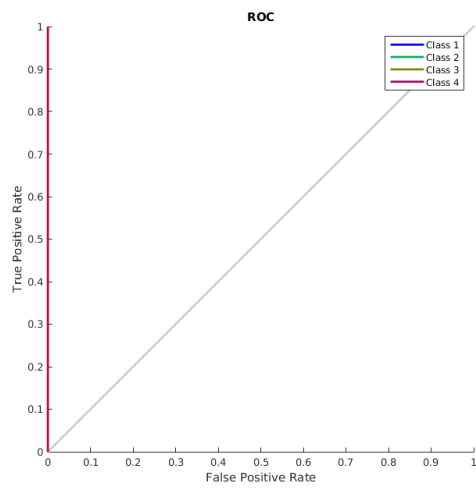
Figura 6.35: Curvas ROC do método C-STIP da base de dados YouTube utilizando a classificação de redes neurais de reconhecimento de padrões.



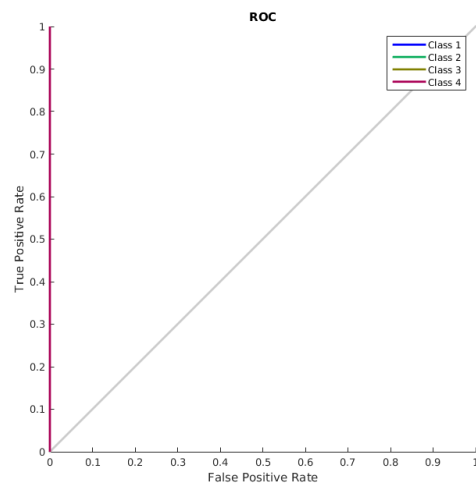
(a) Cenário de treinamento CT_1



(b) Cenário de treinamento CT_2



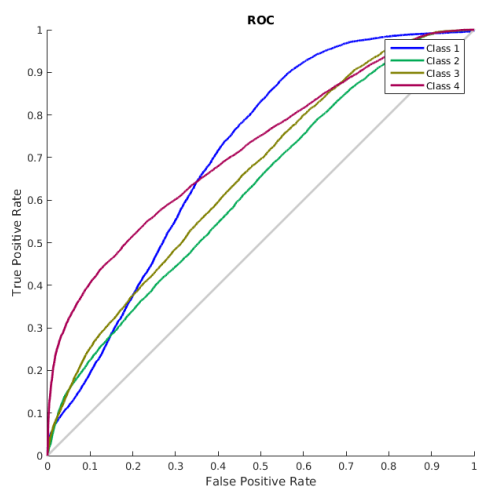
(c) Cenário de treinamento CT_5



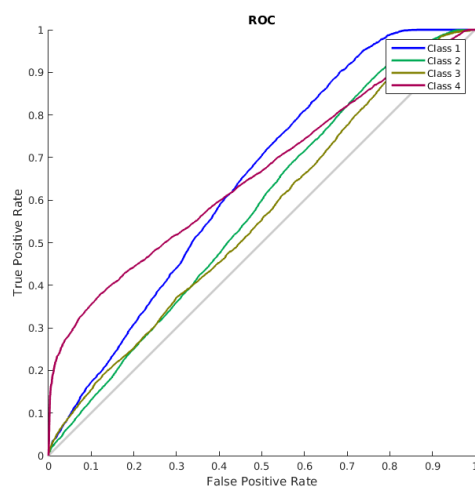
(d) Cenário de treinamento CT_6

Figura 6.36: Curvas ROC do método V-STIP da base de dados YouTube utilizando a classificação de redes neurais de reconhecimento de padrões.

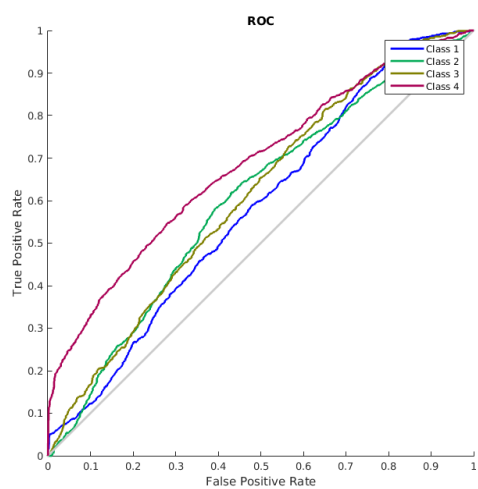
6.2.2 Curvas ROC da Rede Neural de Ajustes de Funções



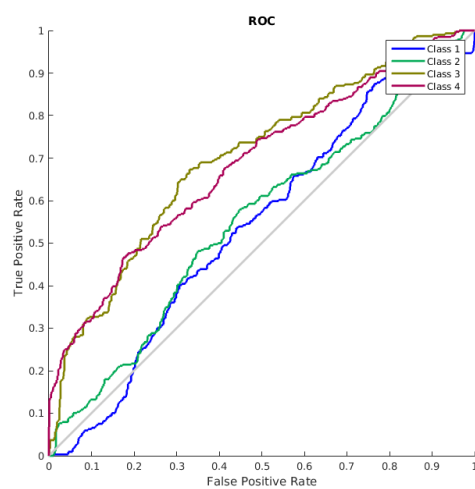
(a) Cenário de treinamento CT_1



(b) Cenário de treinamento CT_2

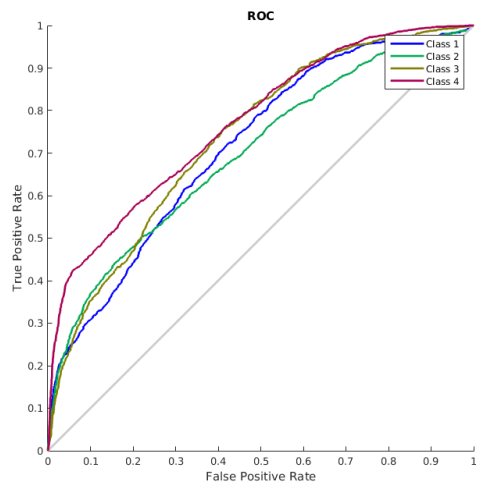


(c) Cenário de treinamento CT_5

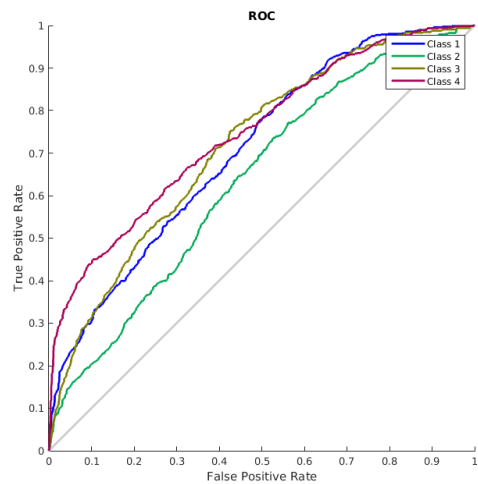


(d) Cenário de treinamento CT_6

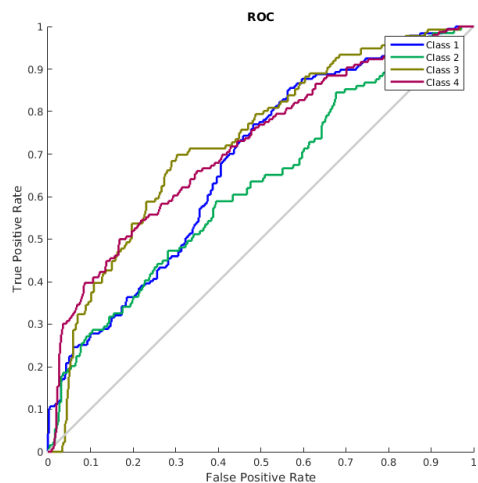
Figura 6.37: Curvas ROC do método C-STIP da base de dados KTH utilizando a classificação de redes neurais de ajustes de funções.



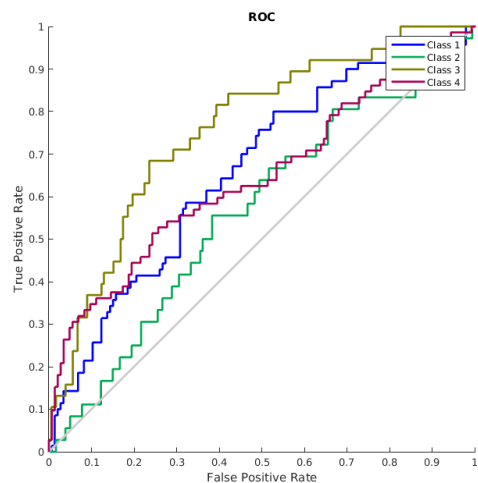
(a) Cenário de treinamento CT_1



(b) Cenário de treinamento CT_2

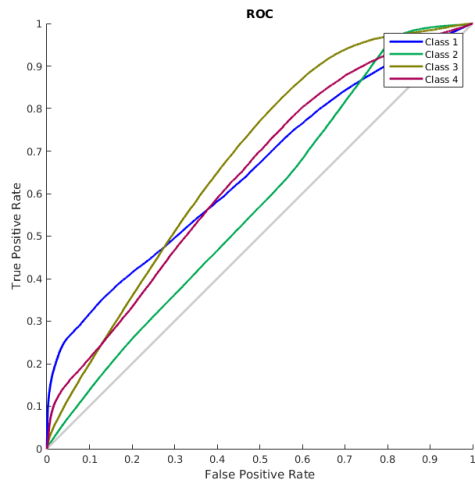


(c) Cenário de treinamento CT_5

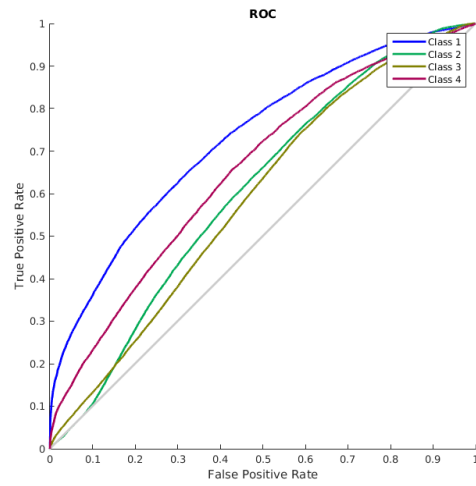


(d) Cenário de treinamento CT_6

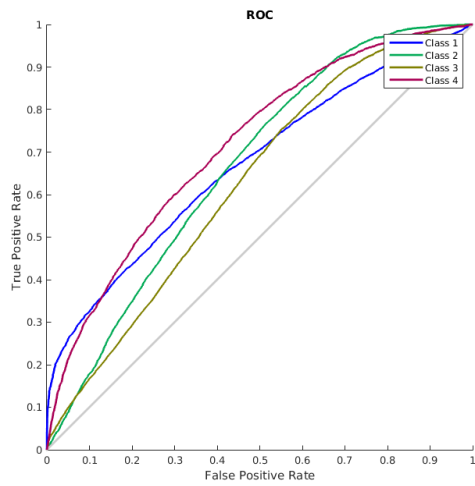
Figura 6.38: Curvas ROC do método V-STIP da base de dados KTH utilizando a classificação de redes neurais de ajustes de funções.



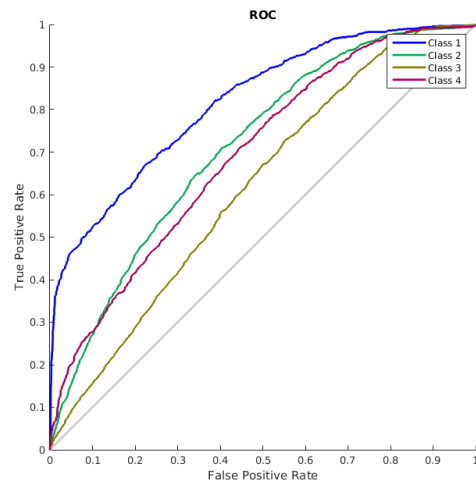
(a) Cenário de treinamento CT_1



(b) Cenário de treinamento CT_2

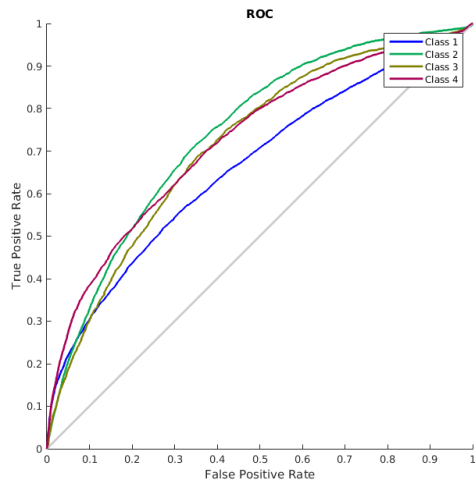


(c) Cenário de treinamento CT_5

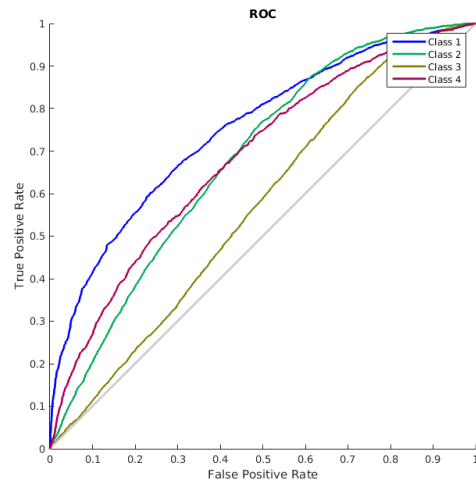


(d) Cenário de treinamento CT_6

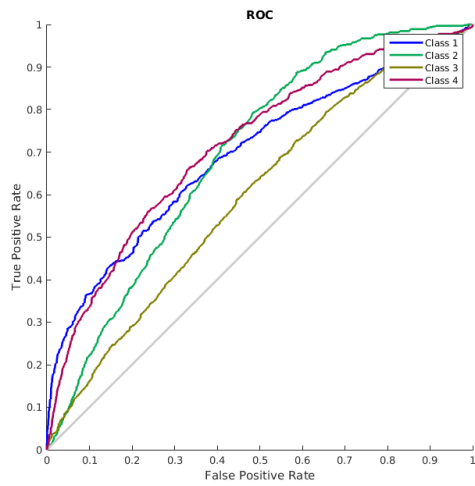
Figura 6.39: Curvas ROC do método C-STIP da base de dados UCF101 utilizando a classificação de redes neurais de ajustes de funções.



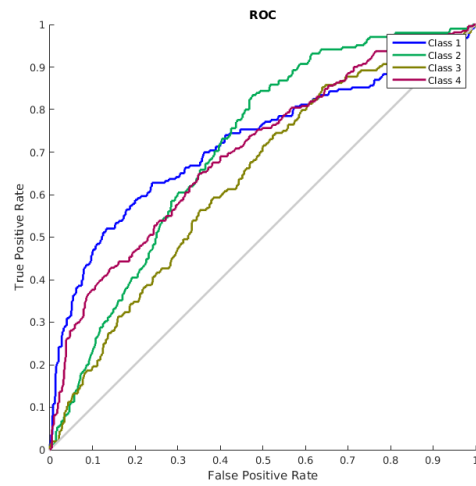
(a) Cenário de treinamento CT_1



(b) Cenário de treinamento CT_2

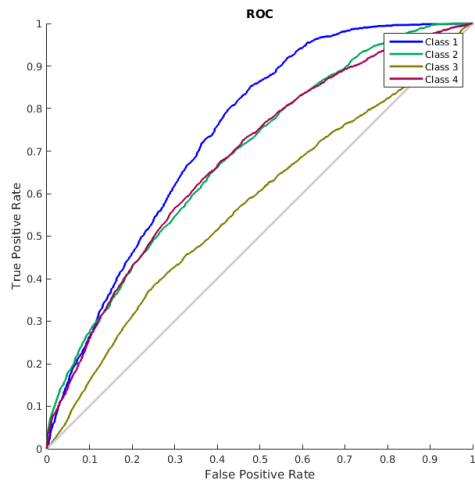


(c) Cenário de treinamento CT_5

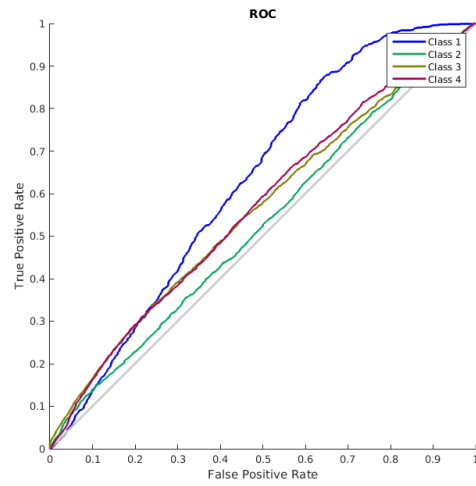


(d) Cenário de treinamento CT_6

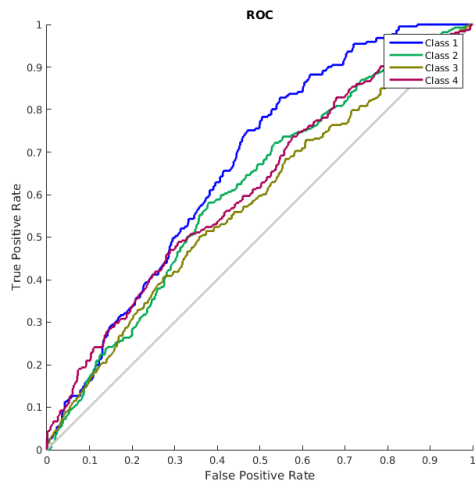
Figura 6.40: Curvas ROC do método V-STIP da base de dados UCF101 utilizando a classificação de redes neurais de ajustes de funções.



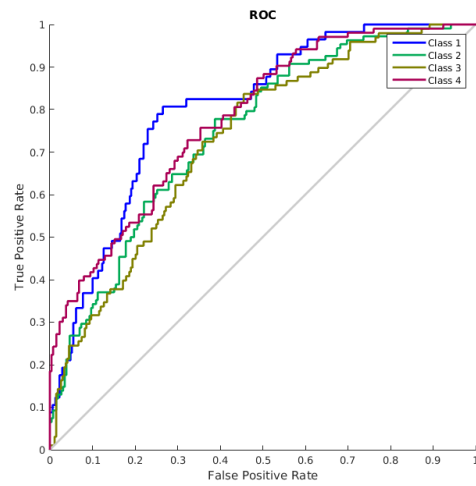
(a) Cenário de treinamento CT_1



(b) Cenário de treinamento CT_2

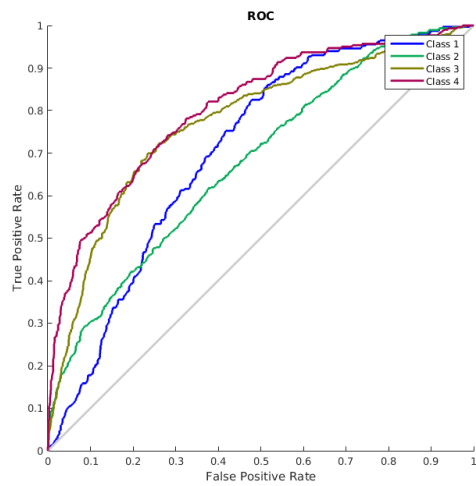


(c) Cenário de treinamento CT_5

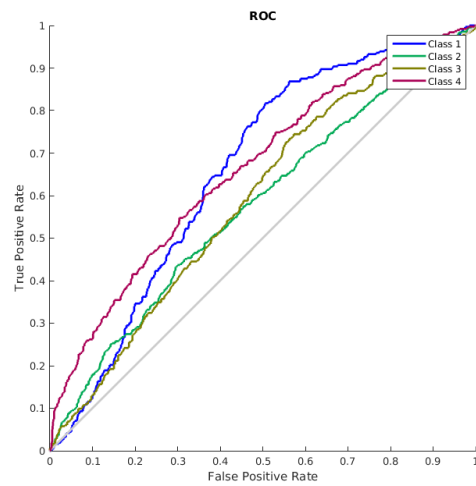


(d) Cenário de treinamento CT_6

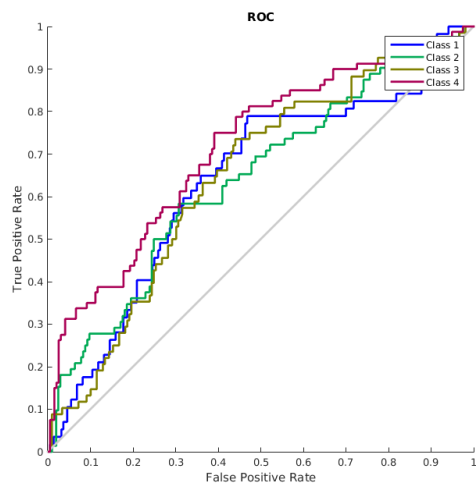
Figura 6.41: Curvas ROC do método C-STIP da base de dados Weizmann utilizando a classificação de redes neurais de ajustes de funções.



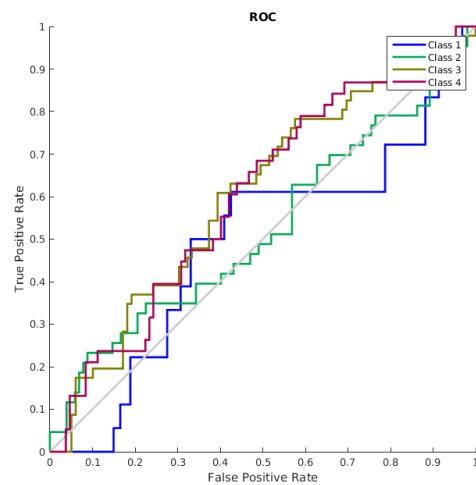
(a) Cenário de treinamento CT_1



(b) Cenário de treinamento CT_2

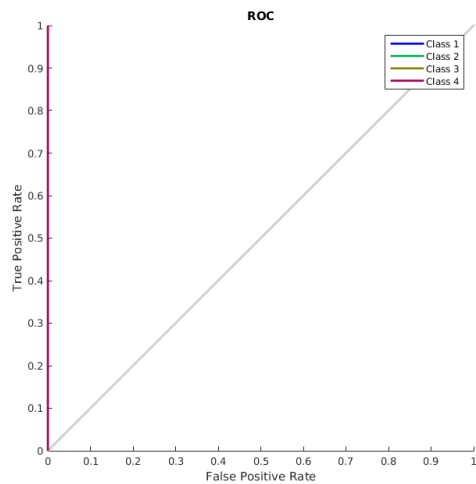


(c) Cenário de treinamento CT_5

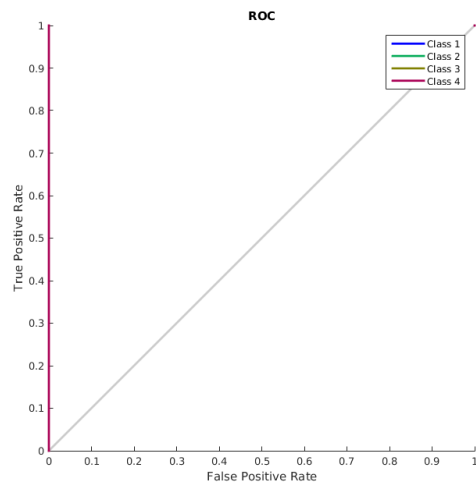


(d) Cenário de treinamento CT_6

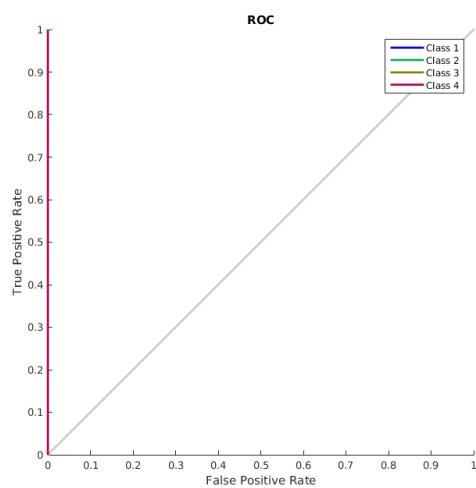
Figura 6.42: Curvas ROC do método V-STIP da base de dados Weizmann utilizando a classificação de redes neurais de ajustes de funções.



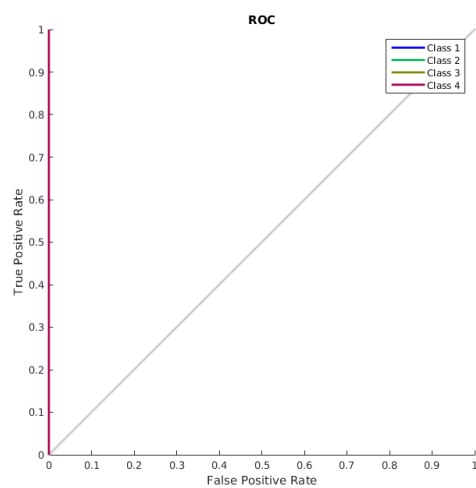
(a) Cenário de treinamento CT_1



(b) Cenário de treinamento CT_2

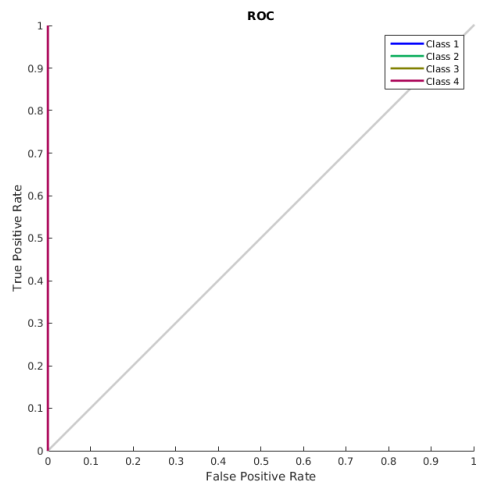


(c) Cenário de treinamento CT_5

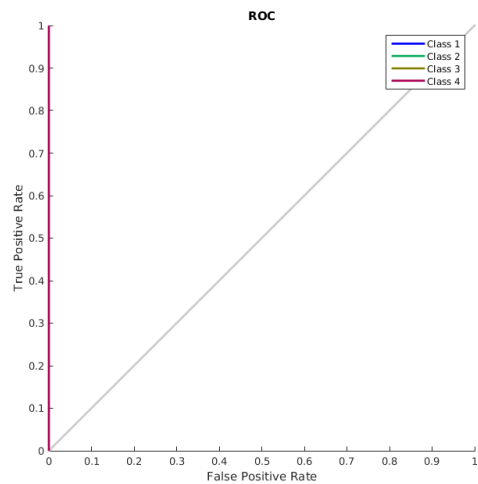


(d) Cenário de treinamento CT_6

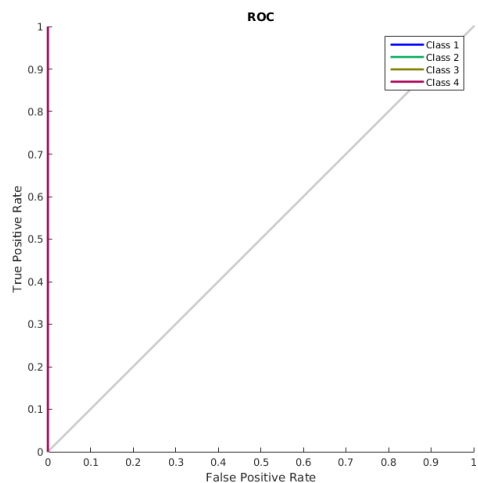
Figura 6.43: Curvas ROC do método C-STIP da base de dados YouTube utilizando a classificação de redes neurais de ajustes de funções.



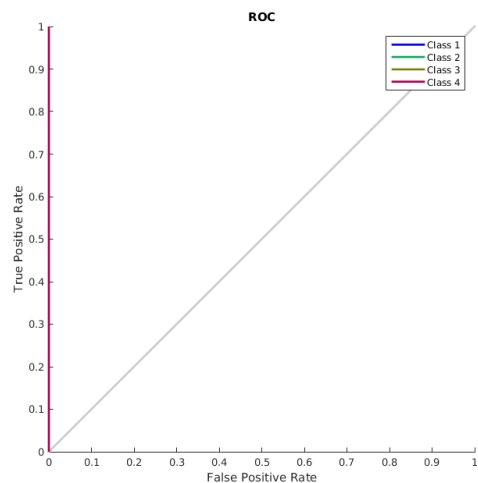
(a) Cenário de treinamento CT_1



(b) Cenário de treinamento CT_2



(c) Cenário de treinamento CT_5

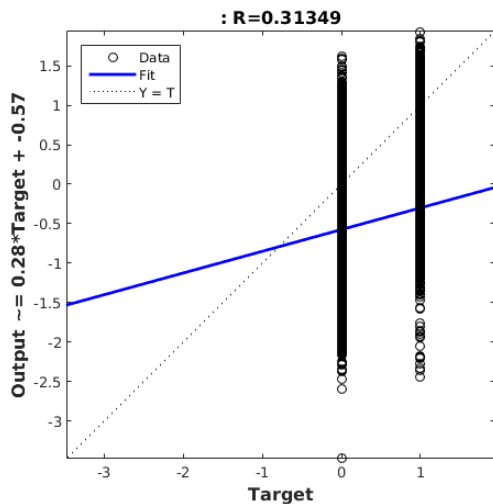


(d) Cenário de treinamento CT_6

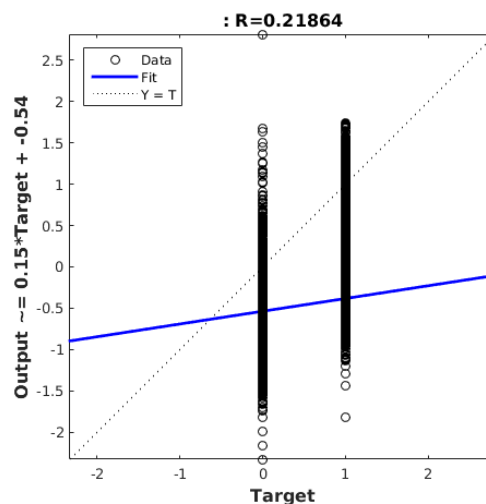
Figura 6.44: Curvas ROC do método V-STIP da base de dados YouTube utilizando a classificação de redes neurais de ajustes de funções.

6.3 Curvas R

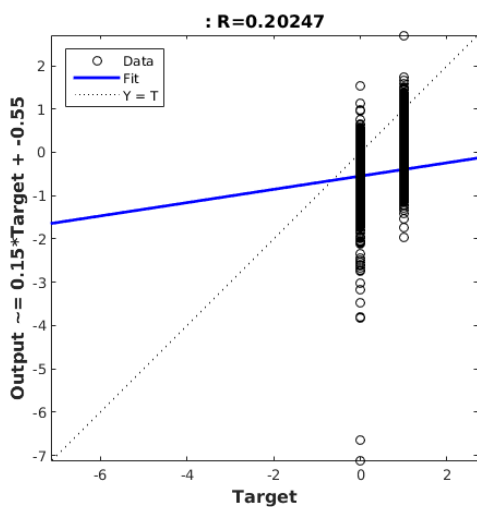
Além das curvas ROC, a rede neural de ajuste de funções possui gráficos R que são mostrados seguidamente.



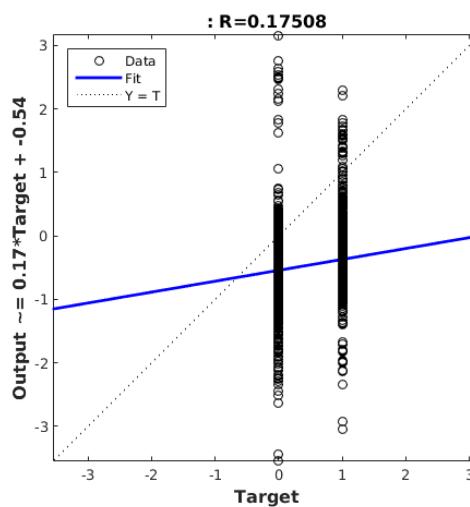
(a) Cenário de treinamento CT_1



(b) Cenário de treinamento CT_2

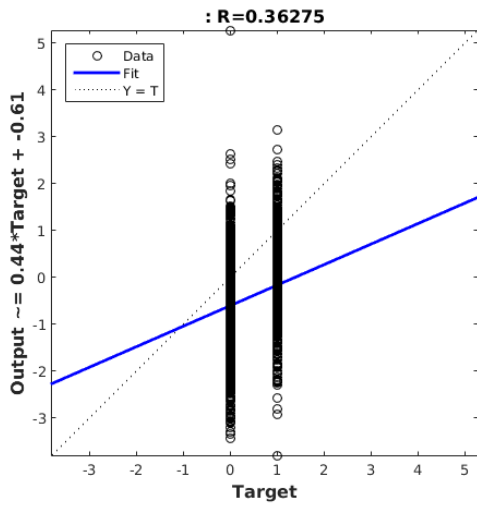


(c) Cenário de treinamento CT_5

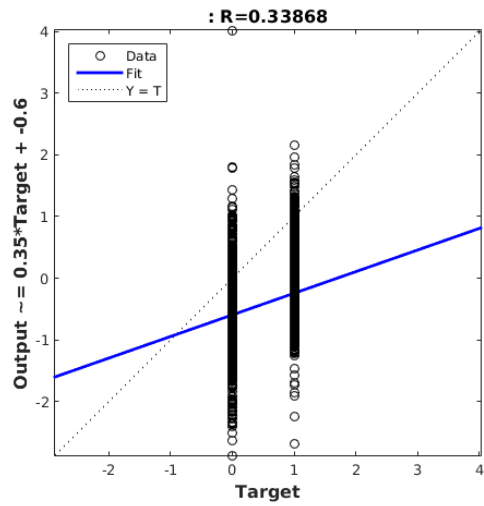


(d) Cenário de treinamento CT_6

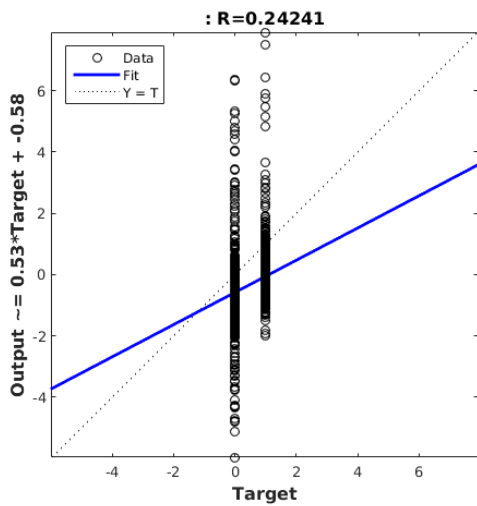
Figura 6.45: Curvas R do método C-STIP da base de dados KTH utilizando a classificação de redes neurais de ajustes de funções.



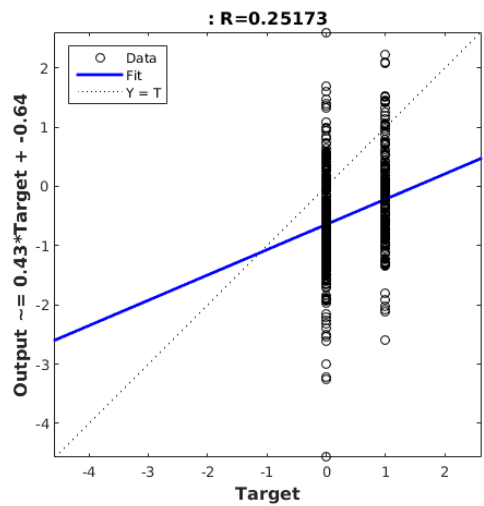
(a) Cenário de treinamento CT_1



(b) Cenário de treinamento CT_2

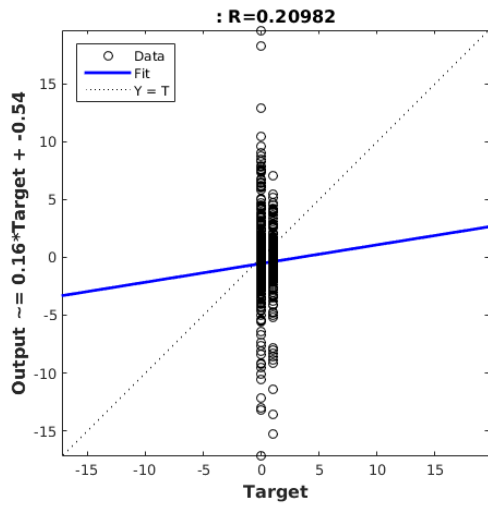


(c) Cenário de treinamento CT_5

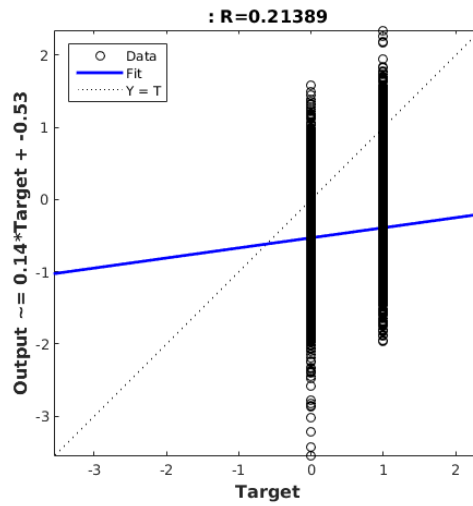


(d) Cenário de treinamento CT_6

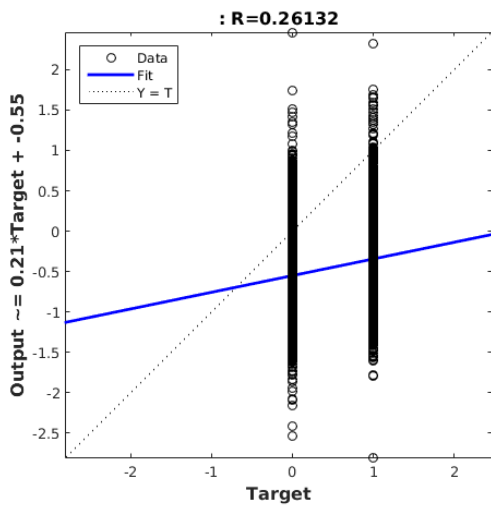
Figura 6.46: Curvas R do método V-STIP da base de dados KTH utilizando a classificação de redes neurais de ajustes de funções.



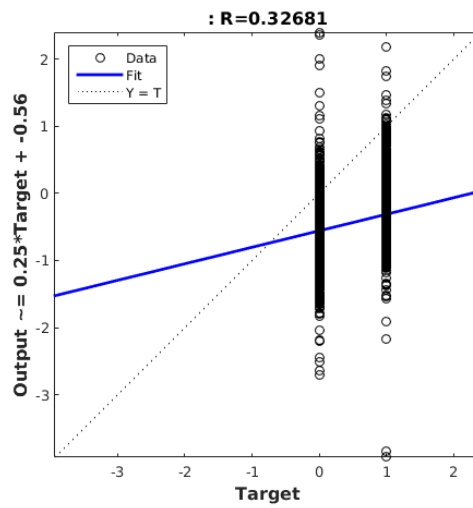
(a) Cenário de treinamento CT_1



(b) Cenário de treinamento CT_2

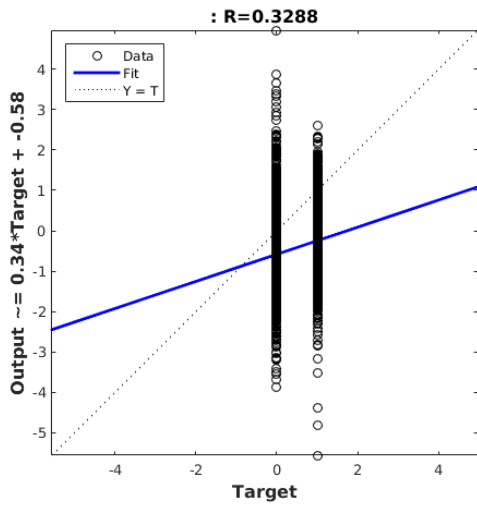


(c) Cenário de treinamento CT_5

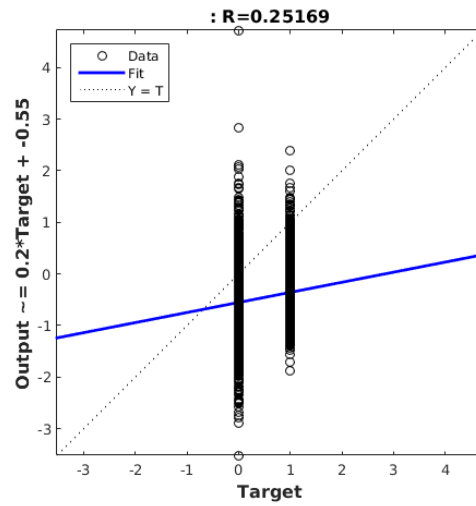


(d) Cenário de treinamento CT_6

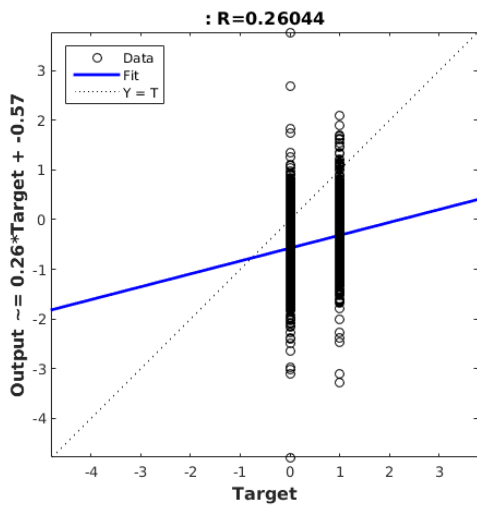
Figura 6.47: Curvas R do método C-STIP da base de dados UCF101 utilizando a classificação de redes neurais de ajustes de funções.



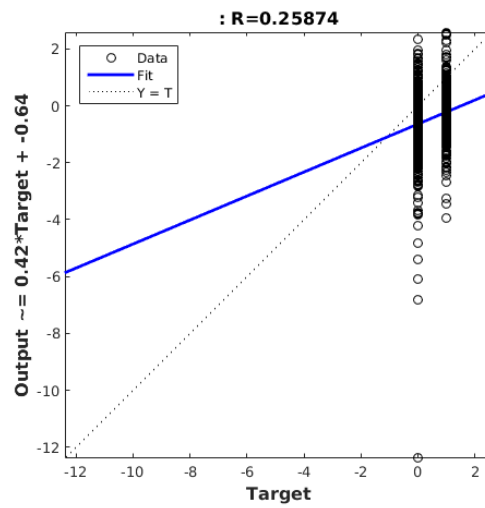
(a) Cenário de treinamento CT_1



(b) Cenário de treinamento CT_2

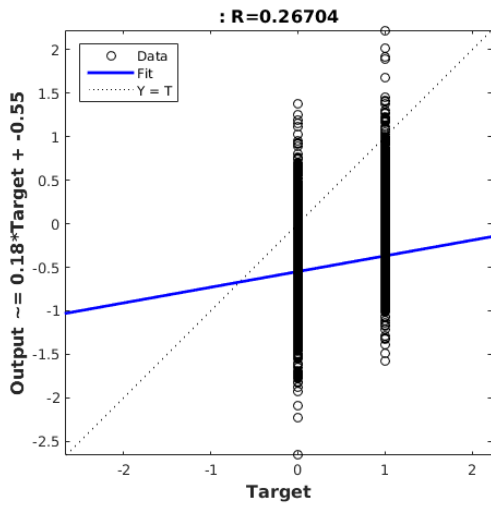


(c) Cenário de treinamento CT_5

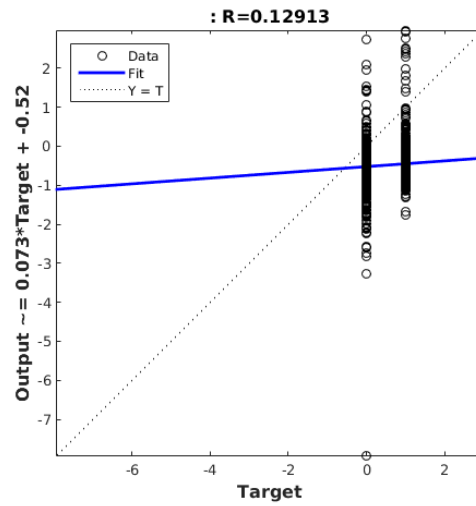


(d) Cenário de treinamento CT_6

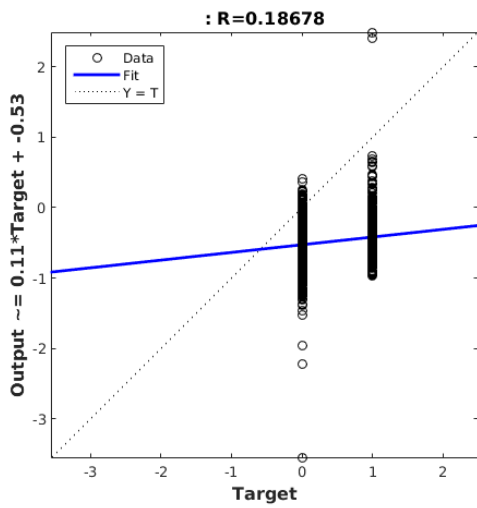
Figura 6.48: Curvas R do método V-STIP da base de dados UCF101 utilizando a classificação de redes neurais de ajustes de funções.



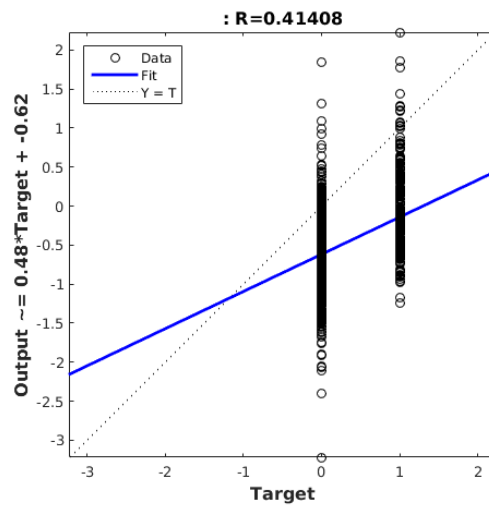
(a) Cenário de treinamento CT_1



(b) Cenário de treinamento CT_2

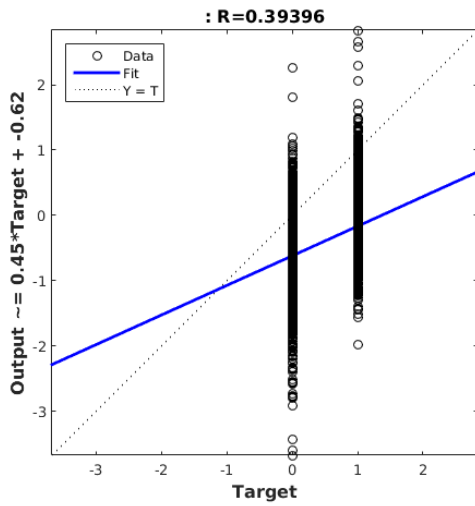


(c) Cenário de treinamento CT_5

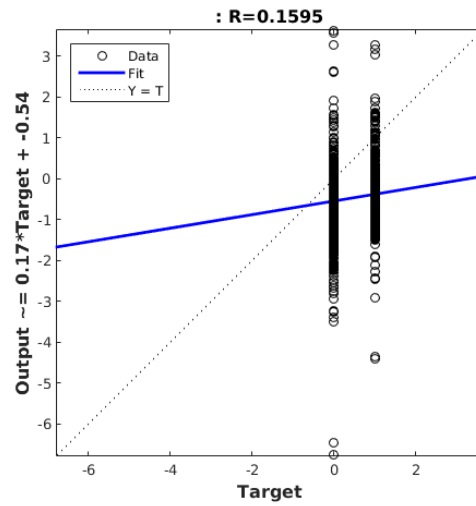


(d) Cenário de treinamento CT_6

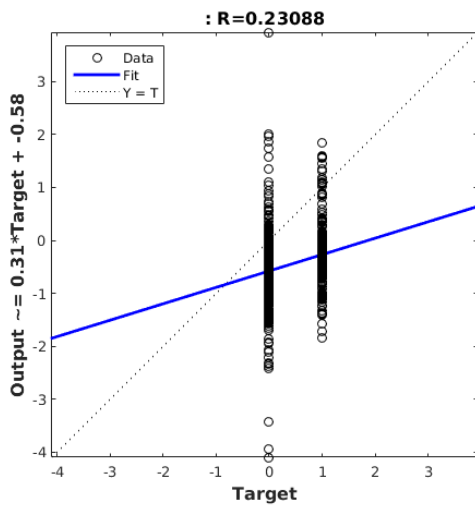
Figura 6.49: Curvas R do método C-STIP da base de dados Weizmann utilizando a classificação de redes neurais de ajustes de funções.



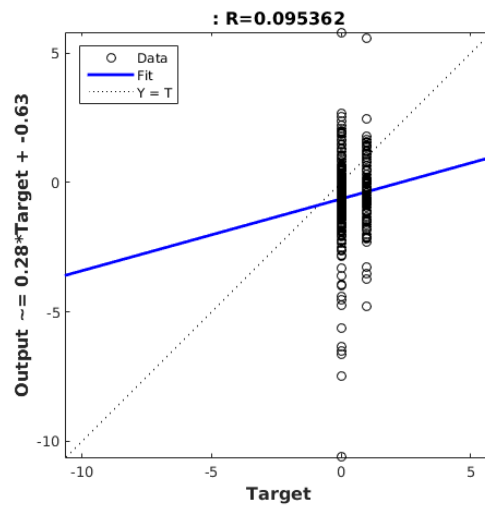
(a) Cenário de treinamento CT_1



(b) Cenário de treinamento CT_2

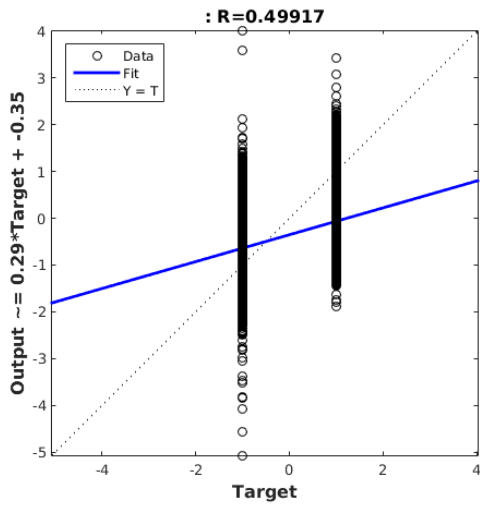


(c) Cenário de treinamento CT_5

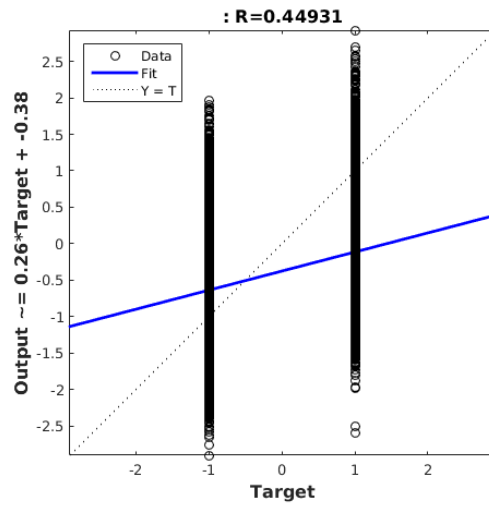


(d) Cenário de treinamento CT_6

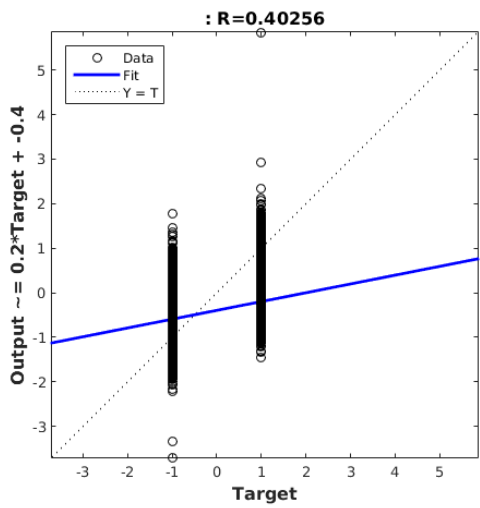
Figura 6.50: Curvas R do método V-STIP da base de dados Weizmann utilizando a classificação de redes neurais de ajustes de funções.



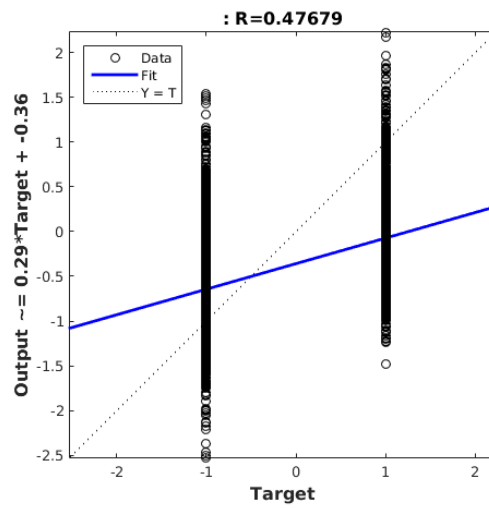
(a) Cenário de treinamento CT_1



(b) Cenário de treinamento CT_2

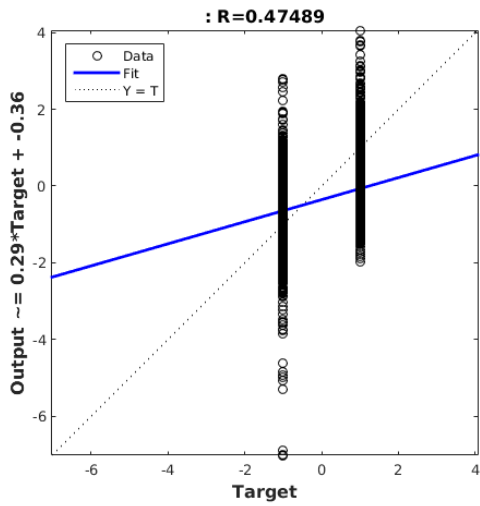


(c) Cenário de treinamento CT_5

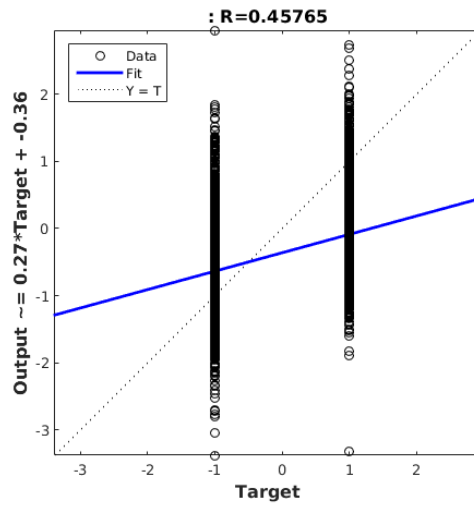


(d) Cenário de treinamento CT_6

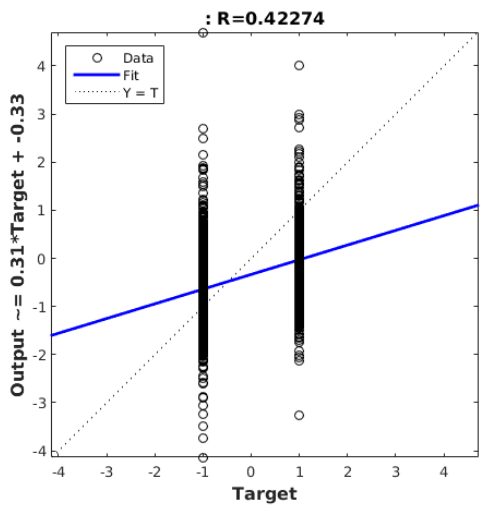
Figura 6.51: Curvas R do método C-STIP da base de dados YouTube utilizando a classificação de redes neurais de ajustes de funções.



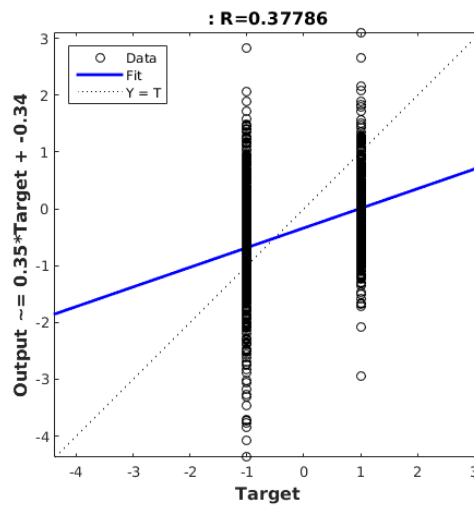
(a) Cenário de treinamento CT_1



(b) Cenário de treinamento CT_2



(c) Cenário de treinamento CT_5



(d) Cenário de treinamento CT_6

Figura 6.52: Curvas R do método V-STIP da base de dados YouTube utilizando a classificação de redes neurais de ajustes de funções.

ANEXO

Neste anexo serão apresentadas as fundamentações teóricas básicas para o uso dos classificadores propostos pela metodologia no Capítulo 4, baseadas em [14].

6.4 Redes Neurais Artificiais

De acordo com [14], as Redes Neurais Artificiais são modelos matemáticos baseados no cérebro humano, tendo como motivação principal o modo como o cérebro processa informações, que é altamente complexo, não-linear e paralelo.

O cérebro realiza processamentos através de seus constituintes estruturais, os neurônios. E esta estrutura é capaz de desenvolver suas próprias regras por meio da experiência acumulada com o tempo. A rede neural é uma máquina que possui a finalidade de modelar a maneira como o cérebro humano trabalha, adquirindo conhecimento pela rede por um processo de aprendizagem com o uso de forças de conexão entre neurônios, os pesos sinápticos, para armazenar esse conhecimento.

O processo de aprendizagem é realizado com a modificação dos pesos sinápticos da rede de forma ordenada.

6.4.1 Perceptrons de Camada Única

O perceptron [63] é construído baseado em um neurônio não-linear, utilizando o modelo de neurônio de *McCulloch-Pitts*[55], apresentado na Figura 6.53. Este modelo é composto por um conjunto de sinapses, cada uma caracterizada por um peso próprio, um somador para somar os sinais de entrada, constituindo um combinador linear, e uma função de ativação para restringir a amplitude de saída do neurônio. O perceptron usa uma função de limiar como função de ativação.

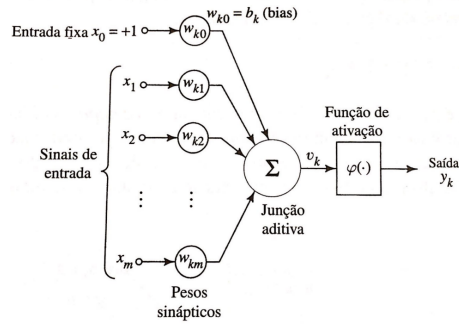


Figura 6.53: Modelo de neurônio não-linear. Retirado de [14].

O *bias* mostrado na figura tem o efeito de aumentar ou diminuir a entrada da função de ativação.

Um neurônio k pode ser descrito a partir do par de Equações 6.1 e 6.2, onde x_1, x_2, \dots, x_m são os sinais de entrada, $w_{k1}, w_{k2}, \dots, w_{km}$ são os pesos sinápticos, u_k é a saída do combinador linear, b_k é o *bias*, $\phi(\cdot)$ é a função de ativação e y_k é o sinal de saída do neurônio.

$$u_k = \sum_{j=1}^m w_{kj}x_j. \quad (6.1)$$

$$y_k = \phi(u_k + b_k). \quad (6.2)$$

O campo local induzido v do perceptron é

$$v = \sum_{i=1}^m w_i x_i + b. \quad (6.3)$$

A finalidade do perceptron é classificar corretamente os sinais de entrada em uma de duas classes ζ_1 ou ζ_2 e para tal, dá-se uma pontuação à ζ_1 caso a saída y for $+1$ e à ζ_2 se a saída $y = -1$. A Figura 6.54 apresenta um mapa de regiões de decisão que é separado por um hiperplano definido pela Equação 6.4 e a fronteira de decisão é dada por $w_1x_1 + w_2x_2 + b = 0$.

$$\sum_{i=1}^m w_i x_i + b = 0. \quad (6.4)$$

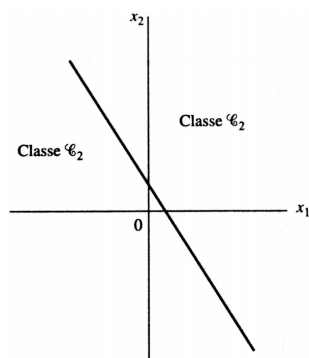


Figura 6.54: Hiperplano como fronteira de decisão para a classificação de duas classes. Retirado de [14].

O pesos sinápticos podem ser adaptados de iteração para iteração, utilizando o algoritmo de convergência do perceptron. Com o vetor de pesos inicial igual a zero, o perceptron é ativado aplicando o vetor de entrada \mathbf{x} e a resposta desejada d . A resposta real do perceptron é então calculada com $y(n) = \text{senal}[\mathbf{w}^T(n)\mathbf{x}(n)]$, onde $\text{senal}(\cdot)$ é a função sinal. Por fim, o vetor de peso do perceptron é atualizado $\mathbf{w}(n+1) = \mathbf{w}(n) + \eta[d(n) - y(n)]\mathbf{x}(n)$, onde η é a constante do parâmetro da taxa de aprendizagem e $d(n) = +1$, se $\mathbf{x}(n)$ pertencer à ζ_1 ou $d(n) = -1$, se $\mathbf{x}(n)$ pertencer à ζ_2 .

6.4.2 Perceptrons de Múltiplas Camadas

As redes de múltiplas camadas alimentadas adiante (*feedforward*) consistem em uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída e o sinal de entrada se propaga para frente, camada por camada. Tais redes são comumente denominadas de Perceptrons de Múltiplas Camadas (*Multilayer Perceptron – MLP*).

Os MLP possuem três características distintas: o modelo de cada neurônio da rede inclui uma função de ativação não-linear; a rede contém uma ou mais camadas de neurônios ocultos, que não são parte da entrada ou da saída e a rede apresenta um alto grau de conectividade, determinado pelas sinapses da rede. Devido à complexidade apresentada pelas características das MLP, o processo de aprendizagem se torna, de maneira resultante, mais abstruso.

O algoritmo de retropropagação (*back-propagation*) [64] viabiliza um método eficiente para o treinamento dos MLPs. A denominação deste algoritmo advém do fato de que as derivadas parciais da função de custo em relação aos parâmetros da rede, como pesos sinápticos e *bias*, são determinados por retropropagação do cálculo dos sinais de erro pelos neurônios de saída através da rede, camada por camada.

6.5 Máquinas de Vetores de Suporte

As máquinas de vetores de suporte (SVM), bem como os MLPs, são redes alimentadas adiante e também podem ser usadas para classificação de padrões e regressões lineares.

Considerando um vetor de entrada \mathbf{x} e a resposta desejada d como uma amostra do treinamento, a equação de uma decisão na forma de hiperplano que realiza esta separação se dá por

$$\mathbf{w}^T \mathbf{x} + b = 0 \quad (6.5)$$

onde \mathbf{w} é um vetor peso ajustável e b o *bias*, pode-se dizer que

$$\begin{aligned} \mathbf{w}^T \mathbf{x}_i + b &\geq 0, d_i = +1, \\ \mathbf{w}^T \mathbf{x}_i + b &< 0, d_i = -1. \end{aligned} \quad (6.6)$$

Para um dado vetor peso de \mathbf{w} e *bias* b , a separação entre o hiperplano e o ponto de dado mais próximo é chamado de margem de separação (ρ). A ideia principal do SVM é encontrar o hiperplano em que a margem de separação ρ é máxima. A Figura 6.55 apresenta um hiperplano ótimo para uma entrada bidimensional.

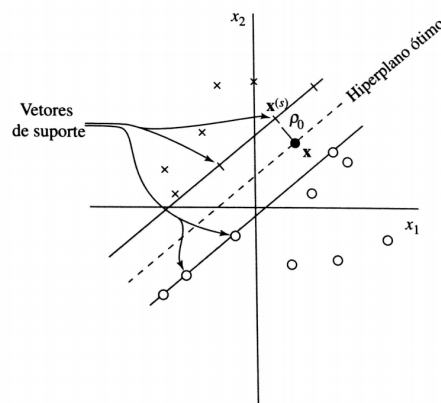


Figura 6.55: Hiperplano ótimo. Retirado de [14].