

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

MARIELLA BERGER ANDRADE

**SISTEMA DE RASTREAMENTO VISUAL DE OBJETOS
BASEADO EM MOVIMENTOS OCULARES SACÁDICOS**

VITÓRIA

2015

MARIELLA BERGER ANDRADE

**SISTEMA DE RASTREAMENTO VISUAL DE OBJETOS
BASEADO EM MOVIMENTOS OCULARES SACÁDICOS**

Tese apresentada ao Programa de Pós-Graduação em
Informática do Centro Tecnológico da Universidade
Federal do Espírito Santo, como requisito parcial
para obtenção do Grau de Doutora em Ciência da
Computação.

VITÓRIA

2015

MARIELLA BERGER ANDRADE

**SISTEMA DE RASTREAMENTO VISUAL DE OBJETOS
BASEADO EM MOVIMENTOS OCULARES SACÁDICOS**

COMISSÃO EXAMINADORA

Prof. Ph.D. Alberto Ferreira De Souza

Universidade Federal do Espírito Santo

Orientador

Prof. Ph.D. Thiago Oliveira dos Santos

Universidade Federal do Espírito Santo

Co-orientador

Profa. Dra. Claudine Santos Badue Gonçalves

Universidade Federal do Espírito Santo

Prof. Ph.D. Edilson de Aguiar

Universidade Federal do Espírito Santo

Prof. Ph.D. Evandro Salles

Universidade Federal do Espírito Santo

Prof. Ph.D. Felipe M. G. França

Universidade Federal do Rio de Janeiro

Vitória, 09 de Abril de 2015.

DEDICATÓRIA

À Maria.

AGRADECIMENTOS

Inicialmente a Deus, minha fonte de força, inspiração, esperança e fé. A minha mãe, Maria, uma lutadora que acreditou que a educação seria a melhor forma de mudar nossas vidas. Esta conquista é dela, por ela e para ela! Ao meu irmão, Junior, por ter sonhado este sonho com a nossa mãe. Obrigada por ser sempre uma parte tão presente de mim. Obrigada por ser meu melhor amigo! Ao meu grande amor Jociel, meu grande presente de Deus. Obrigada por me apoiar sempre e por ter estado ao meu lado, segurando minha mão, nestes anos tão atribulados. Ao meu filho João Rodolfo, minha vida! Mesmo sem muito entender, obrigada por ter passado madrugadas no LCAD sem nunca reclamar! Ao Bernardo, que Deus mandou para acalantar nossos corações. A minha irmã Raphaela pelo carinho! Ao meu orientador (professor Dr. Alberto Ferreira De Souza) e ao meu co-orientador (professor Dr. Thiago Oliveira dos Santos) pela confiança, por todo o conhecimento compartilhado e pela amizade. Obrigada também pela oportunidade concedida. Foi um grande prazer poder trabalhar com vocês! Aos meus amigos do LCAD, em especial ao Avelino, Tiago, Lauro, Filipe, Michael, Rômulo, Lucas, Edilson, Claudine, Bruna, Jorcy e Ranick, que muito me ajudaram no desenvolvimento deste trabalho. Foi e é um prazer trabalhar com vocês. Aos meus alunos que tanto me ensinam e que sempre torceram por mim. Ao professor Raul, que sempre acreditou em mim. Ao professor Saulo, um amigo que me incentivou muito a cursar o doutorado. A minha família que tanto amo. Ao DI e ao PPGI, pela oportunidade e confiança. Foi muito bom fazer parte deste começo de curso de doutorado. Ao CNPQ pelo apoio financeiro que me permitiu estudar.

ΕΠΪΓΡΑΦΕ

"Vision is the act of knowing what is where by looking."

Aristóteles

RESUMO

A busca visual é o mecanismo por meio do qual, a partir do conhecimento prévio da imagem de um objeto de interesse, conseguimos encontrá-lo no campo visual se ele estiver nele presente. A região cerebral responsável pela realização da busca visual, realizada através dos movimentos sacádicos dos olhos, é conhecida como Superior Colliculus.

Um sistema computacional de busca visual biologicamente inspirado precisa modelar o sistema biológico de movimentos sacádicos dos olhos, as transformações sofridas pelas imagens captadas pelos olhos em seu caminho para o Superior Colliculus (SC) no cérebro e a resposta dos neurônios do SC para padrões aprendidos anteriormente.

Neste trabalho, apresentamos uma modelagem matemático-computacional de uma arquitetura neural que representa o Superior Colliculus. Esta arquitetura neural é baseada em Generalização Virtual de Memória de Acesso Aleatório em Redes Neurais Sem Peso (*Virtual Generalizing Random Access Memory Weightless Neural Networks – VGRAM WNN*) e no mapeamento log-polar da retina para o Superior Colliculus. Com a nossa implementação desta arquitetura é possível, a partir de pontos de interesse em uma determinada imagem bidimensional previamente treinados, realizar a busca visual por estes pontos em imagens diferentes da treinada. O modelo de busca visual biologicamente inspirado foi incorporado em um sistema automático de rastreamento (*tracking*) de longo prazo de objetos de interesse em vídeo. Nossos resultados experimentais mostram que nosso sistema de rastreamento visual é capaz de lidar com todos os desafios apresentados e se equipara ao estado da arte em rastreamento de objetos.

ABSTRACT

Visual search is the mechanism that involves a scan of the visual field in order to find an object of interest. The brain region responsible for performing the visual search, performed by saccadic eye movements, is the Superior Colliculus.

A computer system for visual search biologically inspired needs to model the saccadic eye movement, the transformation suffered by the images captured by the eyes in the way from the retina to the Superior Colliculus, and the response of the neurons of the Superior Colliculus to patterns of interest in the visual scene.

In this work, we present a biologically inspired long-term object tracking system based on Virtual Generalizing Random Access Memory (VG-RAM) Weightless Neural Networks (WNN). VG-RAM WNN is an effective machine learning technique that offers simple implementation and fast training. Our system models the biological saccadic eye movement, the transformation suffered by the images captured by the eyes from the retina to the Superior Colliculus (SC), and the response of SC neurons to previously seen patterns. We evaluated the performance of our system using a well-known visual tracking database. Our experimental results show that our approach is capable of reliably and efficiently track an object of interest in a video with accuracy equivalent or superior to related work.

SUMÁRIO

1	INTRODUÇÃO.....	18
1.1	MOTIVAÇÃO	21
1.1	OBJETIVOS	22
1.2	CONTRIBUIÇÕES.....	22
1.3	ESTRUTURA DO TRABALHO	25
2	O SISTEMA VISUAL	26
2.1	O OLHO	26
2.2	VIAS VISUAIS.....	28
2.3	O CÓRTEX VISUAL PRIMÁRIO.....	29
2.4	O SUPERIOR COLLICULLUS	31
2.5	MOVIMENTOS OCULARES SACÁDICOS.....	32
3	MODELO MATEMÁTICO-COMPUTACIONAL DE BUSCA VISUAL.....	34
3.1	MODELO DO MAPEAMENTO RETINITÓPICO.....	34
3.2	REDES NEURAIS SEM PESO VG-RAM	38
3.2.1	<i>Redes Neurais Sem Peso VG-RAM Fat-Fast.....</i>	<i>40</i>
3.3	MODELO DO SUPERIOR COLLICULLUS	42
3.4	O MOVIMENTO SACÁDICO DOS OLHOS.....	47
4	RASTREAMENTO DE OBJETOS COM RNSP VG-RAM.....	51
4.1	(RE)LEARNING.....	52
4.2	TRACKING.....	54
4.2.1	<i>Etapas de Tracking.....</i>	<i>54</i>
4.2.2	<i>Estimativa da Escala</i>	<i>55</i>
4.3	DETECTION	58
4.4	VALIDATION	59
5	AVALIAÇÃO EXPERIMENTAL E RESULTADOS.....	61
5.1	EXPERIMENTO <i>DATASET TLD</i>	61
5.1.1	<i>Metodologia.....</i>	<i>61</i>

5.1.2	<i>Métrica</i>	65
5.1.3	<i>Calibração do Sistema</i>	67
5.1.4	<i>Resultados</i>	109
5.2	EXPERIMENTO SIGA-O-LÍDER	118
5.2.1	<i>Metodologia</i>	120
5.2.2	<i>Implementação</i>	123
5.2.3	<i>Resultado</i>	124
5.3	EXPERIMENTO COM O <i>EYE-TRACKER</i>	126
6	CONCLUSÃO	130
6.1	TRABALHOS FUTUROS	131

LISTA DE FIGURAS

Figura 1: Anatomia do olho humano. Corte do olho direito visto de cima. Figura adaptada de [34].	26
Figura 2: Variação do número de células fotorreceptoras das regiões próximas e distantes da fóvea. Figura adaptada de [34].	27
Figura 3: O ponto cego. Figura adaptada de [34].	28
Figura 4: Ilustração do fluxo de informações visuais da retina ao córtex estriado. Figura retirada de http://www.brainworks.uni-freiburg.de/group/wachtler/VisualSystem/	29
Figura 5: Mapa retinotópico do córtex visual primário humano. Figura retirada de [34].	30
Figura 6: Localização do Superior Colliculus (SC) no cérebro humano. Figura retirada de [34].	31
Figura 7: Mudança do padrão de atividade neural do Superior Colliculus durante a realização de movimentos sacádicos. Retirado de [42].	32
Figura 8: Comparação dos movimentos sacádicos entre cenas. Em a) observamos uma cena com distratores semelhantes ao objeto de interesse e em b) observamos uma cena com distratores distintos do objeto de interesse. Fonte: [45].	33
Figura 9: Representação de uma imagem de círculos concêntricos projetada na retina em V1. Figura retirada de [38].	35
Figura 10: Transformada log-polar. Figura retirada de [42].	36
Figura 11: Aplicação da transformação log-polar seguida pelo mapeamento inverso. Figura retirada de http://omni.isr.ist.utl.pt/~alex/Projects/TemplateTracking/logpolar.htm	37
Figura 12: Transformada log-polar empregada pelo Grupo de Pesquisa em Cognição Visual do LCAD. Figura retirada de [42].	37
Figura 13: Visão geral da arquitetura dos neurônios VG-RAM Fat-Fast. Figura retirada de [60].	41
Figura 14: Arquitetura proposta do Superior Colliculus (SC). Figura retirada de [61].	42
Figura 15: O modelo do Superior Colliculus (SC) proposto. Figura retirada de [61].	43
Figura 16: Ilustração da fase de treino. O centro de atenção é inicialmente movido para o centro do objeto a ser aprendido. Em seguida, a camada neural é pintada com a cor de saída esperada por cada neurônio. Neurônios presentes na caixa delimitadora do objeto são considerados ativos e têm a cor diferente de preto associada a eles. Neurônios fora da caixa delimitadora do objeto são considerados não-ativos e têm a cor preta associada a eles. As	

cores são organizadas de forma que a cor da saída de um neurônio diz qual é o deslocamento deste neurônio do centro do objeto de interesse. Figura retirada de [61].	46
Figura 17: Ilustração da fase de teste com o centro de atenção perto do centro do objeto de interesse (panda). Figura retirada de [61].	47
Figura 18: Ilustração da arquitetura usada para determinar o alvo do movimento sacádico. Cada neurônio ativado, $n_{i,j}$, da camada neural contribui com um voto para a provável localização, $V_{i,j}$, do centro do objeto de interesse. Cada voto é ponderado por uma medida de confiança da resposta de um neurônio, $w_{i,j}$. Os votos são armazenados em uma matriz acumuladora, onde as células representam as possíveis localizações espaciais do objeto de interesse. O local mais votado é escolhido como alvo do movimento sacádico. Figura retirada de [61].	48
Figura 19: O deslocamento de um neurônio. O deslocamento $V_{i,j}$ é obtido adicionando o vetor $L_{i,j}(r, \theta)$ ao vetor $-C_{i,j}(r_c, g_c)$. $L_{i,j}(r, \theta)$ corresponde à localização do centro do campo receptivo do neurônio $n_{i,j}$ na camada de entrada. $C_{i,j}(r_c, g_c)$ corresponde ao deslocamento do centro do objeto de interesse em relação ao centro do campo receptivo do neurônio $n_{i,j}$. Figura retirada de [61].	49
Figura 20: Ilustração da acumulação de votos para a possível localização espacial do objeto de interesse. Cada célula no acumulador corresponde à localização indicada pelo vetor de deslocamento $V_{i,j} = L_{i,j}(r, \theta) - C_{i,j}(r_c, g_c)$ de cada neurônio $n_{i,j}$. Figura retirada de [61].	50
Figura 21: Resultados do sistema de rastreamento visual. Em (a), a caixa delimitadora do objeto de interesse é apresentada em vermelho. Em (b), a saída do sistema para um <i>frame</i> diferente é mostrada em verde em conjunto com uma caixa delimitadora anotada manualmente em vermelho. Em (c), o objeto não é visível e o sistema não apresenta nenhuma caixa delimitadora. Figura retirada de [61].	51
Figura 22: Diagrama de bloco do sistema de rastreamento visual. Figura retirada de [61].	52
Figura 23: Ilustração do mecanismo de ajuste automático de escala. Cada posição no vetor acumulador de fatores de escala corresponde a uma escala indicada por $z_{i,j} = C_{i,j}(r_c, g_c) / L_{i,j}(r, \theta) $. Esta posição é incrementada de acordo com o peso correspondente $w_{i,j}$ de $n_{i,j}$. Um filtro gaussiano é utilizado para suavizar os dados no vetor acumulador. Figura retirada de [61].	57
Figura 24: Objetos em diferentes escalas com seus respectivos estados do vetor acumulador de fatores de escala. Figura retirada de [61].	58
Figura 25: Esquema de criação de uma aplicação utilizando o framework MAE. Figura retirada de [53].	65
Figura 26: Ilustração do coeficiente de Jaccard.	66
Figura 27: Gráfico que apresenta as mudanças ocorridas no desempenho das RNSP ao longo do crescimento da dimensão da rede.	69

Figura 28: Gráfico de tendências de desempenho da variação de sinapses em função da média do desempenho das dimensões testadas da RNSP.....	70
Figura 29: Gráfico de tendências de desempenho da variação do desvio padrão da distribuição normal em função da média do desempenho das dimensões testadas da RNSP.....	71
Figura 30: Gráfico de tendências de desempenho da variação do fator de log em função da média de desempenho das dimensões testadas da RNSP.....	72
Figura 31: Gráfico que apresenta as mudanças ocorridas no desempenho da RNSP ao longo do crescimento da dimensão da RNSP.....	73
Figura 32: Gráfico de tendências de desempenho da variação das sinapses em função da média do desempenho das dimensões testadas da RNSP.....	74
Figura 33: Gráfico de tendências de desempenho da variação do desvio padrão da distribuição normal em função da média do desempenho das dimensões testadas da RNSP.....	75
Figura 34: Gráfico de tendências de desempenho da variação do fator de log em função da média do desempenho das dimensões testadas da RNSP.....	76
Figura 35: Gráfico que apresenta os resultados do desempenho da RNSP com neurônios do tipo VG-RAM em função dos limiares utilizado para retreino do sistema.....	78
Figura 36: Gráfico que apresenta os resultados do desempenho da RNSP com neurônios do tipo VG-RAM Fat-Fast em função dos limiares utilizado para retreino do sistema.....	83
Figura 37: Gráfico que apresenta os resultados do desempenho da RNSP com neurônios do tipo VG-RAM em função de s.....	88
Figura 38: Gráfico que apresenta os resultados do desempenho da RNSP com neurônios do tipo VG-RAM Fat-Fast em função de s.....	91
Figura 39: Gráfico que apresenta os resultados do desempenho da RNSP com neurônios do tipo VG-RAM em função de j.....	94
Figura 40: Gráfico que apresenta os resultados do desempenho da RNSP com neurônios do tipo VG-RAM Fat-Fast em função de j.....	96
Figura 41: Gráfico que apresenta os resultados do desempenho da RNSP com neurônios do tipo VG-RAM em função de numPixels.....	99
Figura 42: Gráfico que apresenta os resultados do desempenho da RNSP com neurônios do tipo VG-RAM Fat-Fast em função de numPixels.....	101
Figura 43: Gráfico que apresenta os resultados do desempenho da RNSP com neurônios do tipo VG-RAM em função de m.....	104
Figura 44: Gráfico que apresenta os resultados do desempenho da RNSP com neurônios do tipo VG-RAM Fat-Fast em função de m.....	107

Figura 45: Resumo dos coeficientes de jaccard por frame para todos os vídeos. As caixas representam o valor médio com um desvio padrão para baixo e um para cima e as linhas representam os valores mínimo e máximo alcançados.....	111
Figura 46: Resultados do rastreamento visual dos frames selecionados do vídeo David (da esquerda para a direita, Frame 1, Frame 190, Frame 380, Frame 570 e Frame 761). Figura retirada de [61].....	112
Figura 47: Resultados do rastreamento visual dos frames selecionados do vídeo Jumping (da esquerda para a direita, Frame 1, Frame 78, Frame 156, Frame 234 e Frame 313). Figura retirada de [61].....	112
Figura 48: Resultados do rastreamento visual dos frames selecionados do vídeo Pedestrian1 (da esquerda para a direita, Frame 1, Frame 35, Frame 70, Frame 105 e Frame 140). Figura retirada de [61].....	113
Figura 49: Resultados do rastreamento visual dos frames selecionados do vídeo Pedestrian2 (da esquerda para a direita, Frame 1, Frame 84, Frame 169, Frame 253 e Frame 338). Figura retirada de [61].....	113
Figura 50: Resultados do rastreamento visual dos frames selecionados do vídeo Pedestrian3 (da esquerda para a direita, Frame 1, Frame 46, Frame 92, Frame 138 e Frame 184). Figura retirada de [61].....	113
Figura 51: Resultados do rastreamento visual dos frames selecionados do vídeo Car (da esquerda para a direita, Frame 1, Frame 236, Frame 472, Frame 708 e Frame 945). Figura retirada de [61].....	114
Figura 52: Resultados do rastreamento visual dos frames selecionados do vídeo Motocross (da esquerda para a direita, Frame 1, Frame 666, Frame 1332, Frame 1998 e Frame 2665). Figura retirada de [61].....	114
Figura 53: Resultados do rastreamento visual dos frames selecionados do vídeo Volkswagen (da esquerda para a direita, Frame 1, Frame 2144, Frame 4288, Frame 6432 e Frame 8576). Figura retirada de [61].....	114
Figura 54: Resultados do rastreamento visual dos frames selecionados do vídeo Carchase (da esquerda para a direita, Frame 1, Frame 2482, Frame 4964, Frame 7446 e Frame 9928). Figura retirada de [61].....	115
Figura 55: Resultados do rastreamento visual dos frames selecionados do vídeo Panda (da esquerda para a direita, Frame 1, Frame 750, Frame 1500, Frame 2250 e Frame 3000). Figura retirada de [61].....	115
Figura 56: Anel viário da UFES.....	119
Figura 57: Plataforma IARA.....	120
Figura 58: Recursos computacionais da plataforma IARA.....	121

Figura 59: Sensores instalados na IARA.....	121
Figura 60: Janela utilizada para demarcação da caixa delimitadora do objeto de interesse a ser rastreado.....	125
Figura 61: Resultado do Sistema de Rastreamento Visual para um frame subsequente ao treinado.	125
Figura 62: Mapa de navegação da IARA. O retângulo amarelo é o próximo objetivo a ser alcançado pelo IARA. O retângulo branco é o IARA representado no mapa 2D.	126
Figura 63: Eye-tracker SMI.....	127
Figura 64: Resultado obtido com o eye-tracker.....	128
Figura 65: Resultado obtido com o Sistema de Rastreamento Visual.....	128
Figura 66: Gráfico com os resultados obtidos com o eye-tracker.	129
Figura 67: Gráfico com os resultados obtidos com o sistema de rastreamento visual proposto.	129

LISTA DE TABELAS

Tabela 1: Tabela-verdade de um neurônio da RNSP VG-RAM. Retirada de [53].	39
Tabela 2: Exemplos de frames dos vídeos do dataset TLD.	63
Tabela 3: Propriedades do dataset TLD.	64
Tabela 4: Conjunto de valores testados para cada parâmetro da rede.	67
Tabela 5: Mudanças ocorridas no desempenho da RNSP ao longo do crescimento da dimensão da rede.	69
Tabela 6: Mudanças ocorridas no desempenho da RNSP ao longo do crescimento da dimensão da rede.	73
Tabela 7: Conjunto de valores testados para cada parâmetro do sistema.	77
Tabela 8: Resultados obtidos com neurônios do tipo VG-RAM com limiar de retreino igual a 20.	79
Tabela 9: Resultados obtidos com neurônios do tipo VG-RAM com limiar de retreino igual a 30.	79
Tabela 10: Resultados obtidos com neurônios do tipo VG-RAM com limiar de retreino igual a 35.	79
Tabela 11: Resultados obtidos com neurônios do tipo VG-RAM com limiar de retreino igual a 40.	79
Tabela 12: Resultados obtidos com neurônios do tipo VG-RAM com limiar de retreino igual a 45.	80
Tabela 13: Resultados obtidos com neurônios do tipo VG-RAM com limiar de retreino igual a 50.	80
Tabela 14: Resultados obtidos com neurônios do tipo VG-RAM com limiar de retreino igual a 55.	80
Tabela 15: Resultados obtidos com neurônios do tipo VG-RAM com limiar de retreino igual a 60.	81
Tabela 16: Resultados obtidos com neurônios do tipo VG-RAM com limiar de retreino igual a 65.	81
Tabela 17: Resultados obtidos com neurônios do tipo VG-RAM com limiar de retreino igual a 70.	81

Tabela 18: Resultados obtidos com neurônios do tipo VG-RAM com limiar de retreino igual a 80.	82
Tabela 19: Resultados obtidos com neurônios do tipo VG-RAM com limiar de retreino igual a 90.	82
Tabela 20: Resultados obtidos com neurônios do tipo VG-RAM com limiar de retreino igual a 100.	82
Tabela 21: Resultados obtidos com neurônios do tipo VG-RAM <i>fat-fast</i> com limiar de retreino igual a 20.	83
Tabela 22: Resultados obtidos com neurônios do tipo VG-RAM <i>fat-fast</i> com limiar de retreino igual a 30.	84
Tabela 23: Resultados obtidos com neurônios do tipo VG-RAM <i>fat-fast</i> com limiar de retreino igual a 35.	84
Tabela 24: Resultados obtidos com neurônios do tipo VG-RAM <i>fat-fast</i> com limiar de retreino igual a 40.	84
Tabela 25: Resultados obtidos com neurônios do tipo VG-RAM <i>fat-fast</i> com limiar de retreino igual a 45.	84
Tabela 26: Resultados obtidos com neurônios do tipo VG-RAM <i>fat-fast</i> com limiar de retreino igual a 50.	85
Tabela 27: Resultados obtidos com neurônios do tipo VG-RAM <i>fat-fast</i> com limiar de retreino igual a 55.	85
Tabela 28: Resultados obtidos com neurônios do tipo VG-RAM <i>fat-fast</i> com limiar de retreino igual a 60.	85
Tabela 29: Resultados obtidos com neurônios do tipo VG-RAM <i>fat-fast</i> com limiar de retreino igual a 65.	86
Tabela 30: Resultados obtidos com neurônios do tipo VG-RAM <i>fat-fast</i> com limiar de retreino igual a 70.	86
Tabela 31: Resultados obtidos com neurônios do tipo VG-RAM <i>fat-fast</i> com limiar de retreino igual a 80.	86
Tabela 32: Resultados obtidos com neurônios do tipo VG-RAM <i>fat-fast</i> com limiar de retreino igual a 90.	87
Tabela 33: Resultados obtidos com neurônios do tipo VG-RAM <i>fat-fast</i> com limiar de retreino igual a 100.	87
Tabela 34: Resultados obtidos para neurônios VG-RAM com valor de s igual a 1.	88

Tabela 35: Resultados obtidos para neurônios VG-RAM com valor de s igual a 2.....	89
Tabela 36: Resultados obtidos para neurônios VG-RAM com valor de s igual a 3.....	89
Tabela 37: Resultados obtidos para neurônios VG-RAM com valor de s igual a 4.....	89
Tabela 38: Resultados obtidos para neurônios VG-RAM com valor de s igual a 5.....	90
Tabela 39: Resultados obtidos para neurônios VG-RAM <i>fat-fast</i> com valor de s igual a 1.	91
Tabela 40: Resultados obtidos para neurônios VG-RAM <i>fat-fast</i> com valor de s igual a 2.	91
Tabela 41: Resultados obtidos para neurônios VG-RAM <i>fat-fast</i> com valor de s igual a 3.	92
Tabela 42: Resultados obtidos para neurônios VG-RAM <i>fat-fast</i> com valor de s igual a 4.	92
Tabela 43: Resultados obtidos para neurônios VG-RAM <i>fat-fast</i> com valor de s igual a 5.	92
Tabela 44: Resultados obtidos para neurônios VG-RAM com valor de j igual a 1.	94
Tabela 45: Resultados obtidos para neurônios VG-RAM com valor de j igual a 2.	94
Tabela 46: Resultados obtidos para neurônios VG-RAM com valor de j igual a 3.	95
Tabela 47: Resultados obtidos para neurônios VG-RAM com valor de j igual a 4.	95
Tabela 48: Resultados obtidos para neurônios VG-RAM com valor de j igual a 5.	95
Tabela 49: Resultados obtidos para neurônios VG-RAM <i>fat-fast</i> com valor de j igual a 1.....	96
Tabela 50: Resultados obtidos para neurônios VG-RAM <i>fat-fast</i> com valor de j igual a 2.....	97
Tabela 51: Resultados obtidos para neurônios VG-RAM <i>fat-fast</i> com valor de j igual a 3.....	97
Tabela 52: Resultados obtidos para neurônios VG-RAM <i>fat-fast</i> com valor de j igual a 4.....	97
Tabela 53: Resultados obtidos para neurônios VG-RAM <i>fat-fast</i> com valor de j igual a 5.....	97
Tabela 54: Resultados obtidos para neurônios VG-RAM com valor de <i>numPixels</i> igual a 1..	99
Tabela 55: Resultados obtidos para neurônios VG-RAM com valor de <i>numPixels</i> igual a 2..	99
Tabela 56: Resultados obtidos para neurônios VG-RAM com valor de <i>numPixels</i> igual a 3.	100
Tabela 57: Resultados obtidos para neurônios VG-RAM com valor de <i>numPixels</i> igual a 4.	100
Tabela 58: Resultados obtidos para neurônios VG-RAM com valor de <i>numPixels</i> igual a 5.	100

Tabela 59: Resultados obtidos para neurônios VG-RAM <i>fat-fast</i> com valor de <i>numPixels</i> igual a 1.	101
Tabela 60: Resultados obtidos para neurônios VG-RAM <i>fat-fast</i> com valor de <i>numPixels</i> igual a 2.	102
Tabela 61: Resultados obtidos para neurônios VG-RAM <i>fat-fast</i> com valor de <i>numPixels</i> igual a 3.	102
Tabela 62: Resultados obtidos para neurônios VG-RAM <i>fat-fast</i> com valor de <i>numPixels</i> igual a 4.	102
Tabela 63: Resultados obtidos para neurônios VG-RAM <i>fat-fast</i> com valor de <i>numPixels</i> igual a 5.	102
Tabela 64: Resultados obtidos para neurônios VG-RAM com valor de <i>m</i> igual a 1.	104
Tabela 65: Resultados obtidos para neurônios VG-RAM com valor de <i>m</i> igual a 4.	104
Tabela 66: Resultados obtidos para neurônios VG-RAM com valor de <i>m</i> igual a 8.	105
Tabela 67: Resultados obtidos para neurônios VG-RAM com valor de <i>m</i> igual a 16.	105
Tabela 68: Resultados obtidos para neurônios VG-RAM com valor de <i>m</i> igual a 32.	105
Tabela 69: Resultados obtidos para neurônios VG-RAM com valor de <i>m</i> igual a 64.	106
Tabela 70: Resultados obtidos para neurônios VG-RAM com valor de <i>m</i> igual a 128.	106
Tabela 71: Resultados obtidos para neurônios VG-RAM <i>fat-fast</i> com valor de <i>m</i> igual a 1.	107
Tabela 72: Resultados obtidos para neurônios VG-RAM <i>fat-fast</i> com valor de <i>m</i> igual a 4.	107
Tabela 73: Resultados obtidos para neurônios VG-RAM <i>fat-fast</i> com valor de <i>m</i> igual a 8.	108
Tabela 74: Resultados obtidos para neurônios VG-RAM <i>fat-fast</i> com valor de <i>m</i> igual a 16.	108
Tabela 75: Resultados obtidos para neurônios VG-RAM <i>fat-fast</i> com valor de <i>m</i> igual a 32.	108
Tabela 76: Resultados obtidos para neurônios VG-RAM <i>fat-fast</i> com valor de <i>m</i> igual a 64.	109
Tabela 77: Resultados obtidos para neurônios VG-RAM <i>fat-fast</i> com valor de <i>m</i> igual a 128.	109
Tabela 78: Resultados do TLD e obtidos com o sistema de rastreamento visual utilizando neurônios VG-RAM.	111

Tabela 79: Número de retreinos realizados e número de <i>frames</i> em que o módulo Detection foi ativado.	117
Tabela 80: Resultados obtidos com o sistema de rastreamento visual utilizando neurônios VG-RAM <i>Fat-Fast</i>	118

1 INTRODUÇÃO

Rastreamento visual de objetos de interesse pode ser definido como o processo de identificar um determinado objeto em movimento no campo visual e mantê-lo no foco de atenção ao longo do tempo. Os humanos são capazes de realizar rastreamento visual de objetos de interesse de forma eficiente mesmo na presença de situações desafiadoras, tais como movimento abrupto do objeto, oclusões, mudanças no ponto de vista, mudanças no plano de fundo e mudanças na aparência do objeto. Apesar dos avanços recentes da pesquisa nesta área [1] [2] [3], realizar a mesma tarefa com sistemas de computadores ainda é um desafio, pois algoritmos específicos têm que ser criados para lidar com os vários possíveis cenários [1].

O problema de rastreamento visual pode ser formulado como descrito a seguir. Dada uma caixa delimitadora que define o objeto de interesse no primeiro *frame* de um vídeo, deve-se determinar automaticamente a caixa delimitadora do objeto ou indicar que o objeto não é visível em todos os *frames* seguintes do vídeo. O grande desafio é que a aparência do objeto pode mudar após o *frame* inicial, tornando mais difícil detectá-lo ao longo do tempo. Este desafio é enfatizado no rastreamento visual de objetos de longo prazo, uma vez que aumentam as possibilidades de se ter grandes alterações na aparência do objeto com o passar do tempo. Além disso, o objeto pode sofrer oclusão em algumas partes do vídeo e reaparecer na sequência. Uma vez que os objetos podem reaparecer em locais diferentes dos quais foi visto pela última vez, algoritmos adaptados para rastreamento contínuo não podem ser utilizados e a detecção do objeto é necessária. Técnicas de rastreamento de objetos possuem muitas aplicações práticas: visão de robôs, interação humano-computador, anotação automática de vídeos, vigilância automática, monitoramento de tráfego e navegação de veículos. Para revisões da literatura consulte [1] [2] [3] .

Algoritmos de rastreamento de objetos seguem duas principais abordagens [2]: rastreamento recursivo e rastreamento por detecção. Métodos de rastreamento recursivo estimam o estado atual de um objeto de interesse através da aplicação de uma transformação no estado anterior (baseando-se em medições feitas em

imagens anteriores e na atual), sendo suscetível à acumulação de erros [2]. Como exemplo, Lucas e Kanade [4] propuseram um método para estimar o fluxo óptico em uma janela de *pixels* em torno do objeto de interesse e Comaniciu et al. [5] apresentam um rastreador em tempo real com base na média de deslocamento (*mean shift*). Métodos de rastreamento por detecção estimam o estado do objeto de interesse considerando apenas as medições realizadas na imagem atual (evitando o acúmulo de erros), necessitando do treinamento de um detector de objeto de antemão. Um exemplo deste método é proposto por Mustafa et al. [6], que gera visões sintéticas de um objeto através da aplicação de técnicas de deformação de um único modelo e treinam um detector de objeto com as imagens deformadas. Métodos de rastreamento por detecção adaptativos tiram proveito das duas abordagens, atualizando o detector do objeto em tempo real. Outros exemplos de métodos de rastreamento por detecção são apresentados a seguir.

O método apresentado por Avidan [7] integra uma SVM (*Support Vector Machine*) com um rastreador de objetos baseado em fluxo óptico. A técnica proposta por Collins e Liu [8] trata o rastreamento como um problema de classificação binária entre objeto de interesse e fundo da imagem. Javed et al. [9] empregam uma combinação de modelos discriminativos e geradores a fim de rotular os dados de entrada e usá-los para melhorar um detector de objeto. Ross et al. [10] propõem um sistema que aprende de forma incremental a representação do subespaço, adaptando-o às mudanças na aparência do objeto. Adam et al. [11] propõem FragTrack, um método que usa um modelo de aparência baseado em partes estáticas do objeto de interesse e histogramas. Avidan [12] utiliza a autoaprendizagem para atualizar um classificador. Grabner et al. [13] empregam uma abordagem semi-supervisionada para aprender o objeto. Babenko et al. [14] aplicam *Multiple Instance Learning* (MIL) para o rastreamento de objeto. Stalder et al. [15] dividiram as tarefas de detecção, reconhecimento e rastreamento em três classificadores distintos. Santner et al. [16] propõem PROST, um modelo não-adaptativo em cascata de um rastreador baseado em fluxo óptico e em *online random forest*. Hare et al. [17] propõem Struck, que, a partir do problema de classificação binária, gera a predição da saída do rastreador. Kalal et al. [18]

empregam a técnica de Tracking-Learning-Detection (TLD), em que objetos encontrados por um rastreador baseado em fluxo óptico são usados para treinar um detector de objeto. As atualizações são realizadas somente se o objeto encontrado é semelhante ao inicial. Em contraste com os métodos de rastreamento por detecção adaptativos, a saída do detector de objeto é utilizada apenas para reiniciar o rastreador baseado em fluxo óptico em caso de falha. Recentemente, vários métodos baseados no TLD original foram desenvolvidos [19] [20] [21]. Embora consiga-se bons resultados nos rastreamentos realizados com algoritmos desta classe de métodos, eles ainda estão longe de ter o desempenho alcançado pelos seres humanos durante o rastreamento visual.

Neste trabalho, utilizamos uma estratégia para rastreamento de objetos a longo prazo baseada em Generalização Virtual de Memória de Acesso Aleatório em Redes Neurais Sem Peso (*Virtual Generalizing Random Access Memory Weightless Neural Networks – VGRAM WNN*) [22] [23] para implementar um sistema de rastreamento por detecção adaptativo. VG-RAM WNN é uma rede neural que não armazena pesos nas sinapses e, ao contrário das redes neurais padrão, o conhecimento é mantido nos neurônios. Tem sido demonstrado que este tipo de rede apresenta alto desempenho para uma variedade de aplicações de aprendizagem de máquina, como reconhecimento de faces [24] [25], categorização de textos [26], predição do retorno do investimento em ações da bolsa de valores [27], detecção e reconhecimento de placas de trânsito [28] [29] [30] e rastreamento de objetos [31] [32] [33].

A VG-RAM WNN proposta é inspirada na busca visual realizada pelos mamíferos. A busca visual é o mecanismo por meio do qual, a partir do conhecimento prévio da imagem de um objeto de interesse, conseguimos encontrá-lo no campo visual se ele estiver nele presente. Um sistema de busca visual biologicamente inspirado precisa modelar o sistema biológico de movimentos sacádicos dos olhos, as transformações sofridas pelas imagens captadas pelos olhos em seu caminho para o Superior Colliculus (SC) no cérebro e a resposta dos neurônios do SC para padrões aprendidos anteriormente.

O modelo de busca visual biologicamente inspirado foi incorporado em um sistema de rastreamento de longo prazo que lida com todos os desafios discutidos. Avaliamos o desempenho do sistema de rastreamento de objetos utilizando o banco de dados TLD (disponível em <http://personal.ee.surrey.ac.uk/Personal/Z.Kalal/tld.html>). Um vídeo com um resumo dos resultados alcançados está disponível em <https://www.youtube.com/watch?v=rz5-5IG6yU>. Realizamos, também, um experimento do tipo “siga-o-líder” com o carro autônomo IARA do Laboratório de Computação de Alto Desempenho (LCAD) da Universidade Federal do Espírito Santo (UFES). Um vídeo com este experimento está disponível em <https://www.youtube.com/watch?v=lePu4KskvNk>. Realizamos, por fim, um experimento preliminar com um *eye-tracker*, que permite medir e registrar os movimentos oculares de um indivíduo. Um vídeo deste experimento está disponível em <https://www.youtube.com/watch?v=MIMlxHp4uz0>. Os resultados experimentais mostraram que a abordagem proposta é capaz de rastrear de maneira confiável e eficiente uma grande variedade de objetos.

1.1 Motivação

Cognição pode ser definida como a nossa capacidade de compreender o mundo e as ideias por meio dos nossos sentidos e de nossa memória de experiências passadas. Podemos categorizar grosseiramente nossas diversas habilidades cognitivas de acordo com nossos sentidos, levando a categorias tais como cognição visual, cognição auditiva, ou cognição tátil. Podemos categorizar ainda nossas habilidades cognitivas de acordo com nossa capacidade de compreender, via nossos sentidos e memória do passado, conceitos tais com o de espaço (cognição espacial) ou o de movimento/ação do nosso corpo (cognição motora).

A cognição visual é viabilizada no cérebro humano por uma quantidade enorme de neurônios (dezenas de bilhões de neurônios [34]), tratando-se de uma capacidade extremamente complexa. Neste trabalho, aprofundamos os estudos sobre o modelo matemático-computacional de busca visual da visão humana,

buscando compreender como o sistema visual dos mamíferos viabiliza a cognição visual na busca de objetos de interesse.

1.1 Objetivos

O principal objetivo deste trabalho é apresentar um modelo computacionalmente viável para a tarefa de rastreamento de objetos de interesse a longo prazo em imagens bidimensionais, baseado na neurofisiologia da região cerebral conhecida como Superior Colliculus (SC), responsável pela realização da busca visual do objeto de interesse em uma imagem.

O SC é uma região do sistema nervoso central que se localiza na superfície dorsal do mesencéfalo e desempenha papel fundamental no direcionamento do olhar, a partir dos olhos e dos movimentos da cabeça, integrando múltiplas entradas sensoriais e cognitivas.

1.2 Contribuições

As principais contribuições deste trabalho foram:

- Proposição de um modelo matemático-computacional para a busca visual biologicamente plausível baseado na neurofisiologia da região do Superior Collicullus;
- Implementação de um modelo matemático-computacional baseado em Redes Neurais Sem Peso VG-RAM (*Virtual Generalizing RAM* - VG-RAM) que emula os movimentos oculares de sacada observados na busca visual;

- Ajuste de parâmetros (calibração) do modelo implementado, visando encontrar a configuração de melhor desempenho do sistema de busca visual;
- Implementação de um sistema de rastreamento visual utilizando o sistema de busca visual proposto;
- Ajuste de parâmetros (calibração) do sistema de rastreamento visual visando encontrar a configuração de melhor desempenho;
- Comparação do desempenho do sistema de rastreamento de objetos com uma abordagem considerada estado da arte dentre técnicas computacionais para o rastreamento (*tracking*) de objetos em sequências de vídeo [18];
- Implementação de um módulo de rastreamento visual para o carro autônomo IARA, utilizado para a realização de experimentos do tipo “siga-o-líder”;
- Realização de um experimento preliminar de comparação do desempenho do sistema de rastreamento de objetos com o resultado obtido com um *eye tracker*.

Os artigos publicados no decorrer do doutorado foram:

- M. Berger, A. F. De Souza, T. Oliveira-Santos, E. Aguiar, J. O. Neto. Visual Tracking with VG-RAM Weightless Neural Networks, Neurocomputing (Amsterdam), ISSN 0925-2312, 2015.
- M. Berger, A. Forechi, A. F. De Souza, J. O. Neto, L. Veronese, V. Neves, E. Aguiar, C. Badue. Traffic Sign Recognition with WiSARD and VG-RAM Weightless Neural Networks. Journal of Network and Innovative Computing (JNIC), ISSN 2160-2174, v. 1, pp. 87-98, 2013.

- A. F. De Souza, A. Forechi, F. Mutz, M. Berger, T. O. Santos, C. Badue. Programming a VG-RAM based Neural Network Computer, International Joint Conference on Neural Networks (IJCNN 2014), Beijing (China), pp. 3871-3878, 2014.
- E. Aguiar, L. Veronese, M. Berger, A. F. De Souza, C. Badue, T. O. Santos. Compressing VG-RAM WNN Memory for Lightweight Applications, International Joint Conference on Neural Networks (IJCNN 2014), Beijing (China), pp. 1063-1070, 2014.
- A. F. De Souza, C. Fontana, F. Mutz, T. A. Oliveira, M. Berger, A. Forechi, J. O. Neto, E. Aguiar, C. Badue. Traffic Sign Detection with VG-RAM Weightless Neural Networks, International Joint Conference on Neural Networks (IJCNN 2013), Dallas (Texas), pp. 730-738, 2013.
- V. B. Azevedo, A. F. De Souza, L. Veronese, C. Badue, M. Berger. Real-time Road Surface Mapping Using Stereo Matching, V-Disparity and Machine Learning, International Joint Conference on Neural Networks (IJCNN 2013), Dallas (Texas), pp. 2575-2582, 2013.
- L. Veronese, L. J. Lyrio Junior, F. Mutz, J. O. Neto, V. B. Azevedo, M. Berger, A. F. De Souza, C. Badue. Stereo Matching with VG-RAM Weightless Neural Networks, International Conference on Intelligent Systems Design and Applications (ISDA), Kochi (India), pp. 309-314, 2012.
- M. Berger, A. Forechi, A. F. De Souza, J. O. Neto, L. Veronese, C. Badue. Traffic Sign Recognition with VG-RAM Weightless Neural Networks, International Conference on Intelligent Systems Design and Applications (ISDA), Kochi (India), pp. 315-319, 2012.

1.3 Estrutura do Trabalho

Após esta introdução, este trabalho está organizado da seguinte forma:

- Capítulo 2: apresentamos um estudo sobre a visão biológica e o movimento sacádico dos olhos, inspirações para o sistema de rastreamento visual proposto.
- Capítulo 3: apresentamos o modelo matemático-computacional do sistema de busca visual biologicamente inspirado.
- Capítulo 4: apresentamos o sistema de rastreamento visual biologicamente inspirado baseado em VG-RAM WNN.
- Capítulo 5: apresentamos o ambiente computacional, a metodologia de testes adotada, as métricas utilizadas na avaliação de desempenho do sistema implementado e os resultados experimentais obtidos com o modelo implementado neste trabalho.
- Capítulo 6: concluímos este trabalho detalhando suas contribuições dentro do escopo do trabalho proposto, bem como o que esperamos realizar em trabalhos futuros.

2 O SISTEMA VISUAL

Neste capítulo apresentamos um estudo sobre a visão biológica e o movimento sacádico dos olhos, inspirações para o sistema de rastreamento visual proposto.

2.1 O Olho

O olho humano é um órgão complexo e delicado. O globo ocular, com cerca de 2.5 centímetros de diâmetro, é responsável pela captação da luz refletida pelos objetos.

A luz penetra nos olhos através da córnea, uma membrana de aproximadamente 0.5 milímetros de espessura extremamente sensível. Em seguida, esta luz atravessa a região chamada câmara anterior, uma membrana com cerca de 3 milímetros de espessura. Após atravessar a câmara anterior, a luz chega a lente (ou cristalino), responsável pelo ajuste de foco. Por trás da lente, a luz passa pelo humor vítreo. Por fim, após atravessar o humor vítreo, a luz encontra a retina, onde efetivamente ocorre a detecção da luz pelos olhos. A Figura 1 apresenta a anatomia do olho direito visto de cima.

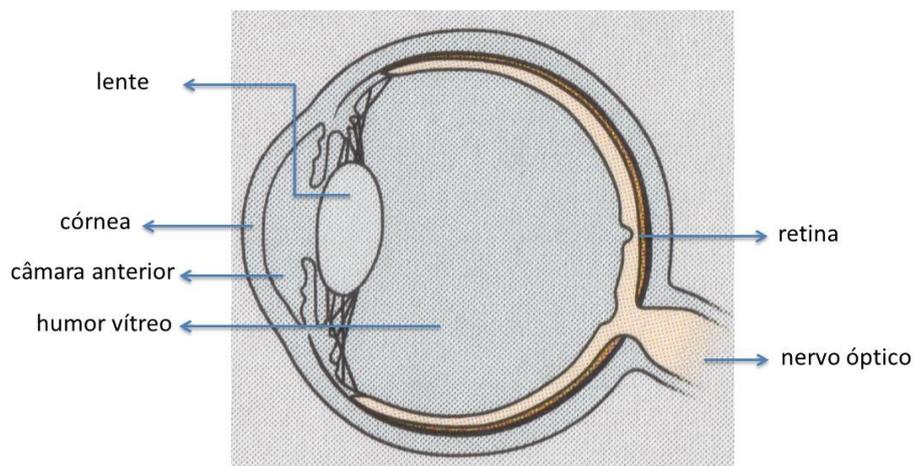


Figura 1: Anatomia do olho humano. Corte do olho direito visto de cima. Figura adaptada de [34].

A retina é um tecido fino, de aproximadamente 0.25 milímetros de espessura, que reveste a parte posterior do globo ocular [35]. Além das células fotorreceptoras sensíveis à luz (cones e bastonetes), a retina possui quatro classes básicas de neurônios: horizontais, bipolares, amácrinas e células ganglionares. Os axônios das células ganglionares que ocupam a superfície da retina formam o nervo óptico, que é responsável por enviar as informações visuais para as demais regiões do cérebro.

Na retina, o número de células fotorreceptoras variam de acordo com a localização espacial (a Figura 2 apresenta a variação do número de células fotorreceptoras das regiões próximas e distantes da fóvea). Por ser uma região completamente avascular (a luz é detectada na fóvea sem nenhuma perda ou dispersão de radiação) e com o maior número de células fotorreceptoras da retina, a fóvea é a região de máxima acuidade visual [36]. Acuidade visual representa a clareza da visão em relação ao foco da retina e a sensibilidade cerebral. Quanto mais distante da fóvea, maior é a área que será coberta por um único campo receptivo, o que acarreta uma queda acentuada na acuidade visual.

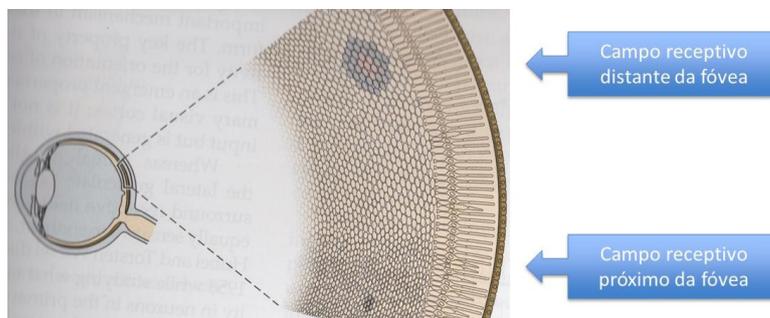


Figura 2: Variação do número de células fotorreceptoras das regiões próximas e distantes da fóvea. Figura adaptada de [34].

O local onde emerge o nervo óptico cria uma região que é chamada de ponto cego, como mostra a Figura 3. Esta região é desprovida de fotorreceptores. Assim, uma imagem projetada nesta região não pode ser percebida.

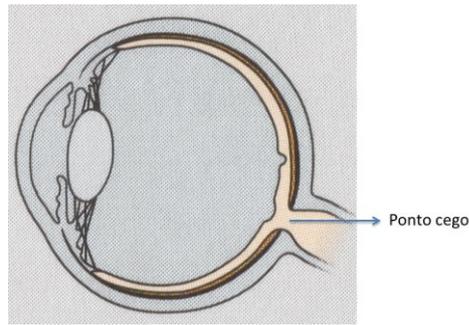


Figura 3: O ponto cego. Figura adaptada de [34].

As informações visuais que chegam à retina são transformadas em pulsos nervosos pelos fotorreceptores e pelas células ganglionares. Após esta transformação, as informações são enviadas para as demais áreas do cérebro, conhecidas como as vias visuais.

2.2 Vias Visuais

Os axônios das células ganglionares presentes na retina do olho humano, através do nervo óptico, projetam as informações visuais para múltiplas áreas do cérebro, dentre elas para o córtex visual primário (V1) [37]. Essas informações atingem uma estrutura chamada quiasma óptico, que inverte o fluxo das informações oriundas da retina para o interior do cérebro. Após esta inversão, as informações visuais chegam a uma região chamada de Núcleo Geniculado Lateral (NGL). A Figura 4 ilustra o fluxo de informações da retina à V1.



Figura 4: Ilustração do fluxo de informações visuais da retina ao córtex estriado. Figura retirada de <http://www.brainworks.uni-freiburg.de/group/wachtler/VisualSystem/>.

As estruturas das camadas do NGL são similares às estruturas ganglionares da retina, possuindo organização similar. Esta organização é chamada de retinotopia ou mapa retinotópico. A organização retinotópica não é linear pois, como dito anteriormente, há uma concentração muito maior de neurônios na região da fóvea do que nas demais regiões da retina, proporcionando uma alta resolução na parte central. Após as informações retinotópicas serem organizadas no NGL, elas são levadas ao córtex visual primário (V1).

2.3 O Córtex Visual Primário

O córtex visual primário, também conhecido como córtex estriado ou V1, é responsável por processar as informações visuais provindas do NGL e por enviar os resultados deste processamento para outras regiões do cérebro, como exemplo, para o Superior Colliculus.

Assim como outras estruturas das vias visuais, o córtex visual primário possui dois hemisférios: o direito (que recebe informações do olho esquerdo) e o esquerdo (que recebe informações do olho direito). Com cerca de 140 milhões de neurônios em cada hemisfério, V1 é também uma das regiões de maior densidade de neurônios do cérebro. A Figura 4 ilustra o mapeamento de uma cena observada pelos olhos e mapeada em V1.

Através de estudos realizados em primatas, verificou-se que o córtex visual primário, assim como o NGL, possui um mapa retinotópico, isto é, áreas do campo visual vizinhas da retina são também vizinhas em V1 [38]. O aspecto mais importante deste mapa é que cerca da metade das projeções da retina sobre o córtex visual primário são provenientes da fóvea e regiões circunvizinhas. A Figura 5 ilustra o mapa retinotópico do cortex visual primário.

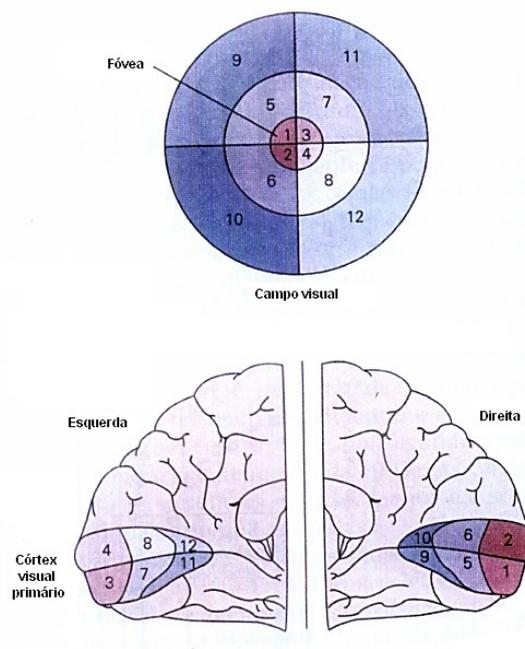


Figura 5: Mapa retinotópico do córtex visual primário humano. Figura retirada de [34].

2.4 O Superior Colliculus

O Superior Colliculus (SC) é uma região do sistema nervoso central que se localiza na superfície dorsal do mesencéfalo e desempenha um papel fundamental no direcionamento do olhar, a partir dos olhos e dos movimentos da cabeça, integrando múltiplas entradas sensoriais e cognitivas. A Figura 6 apresenta em laranja o SC na região do mesencéfalo.

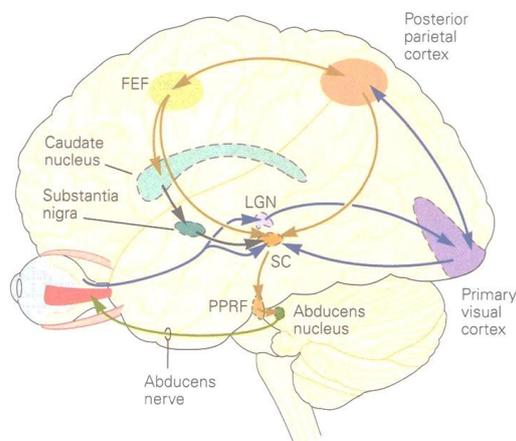


Figura 6: Localização do Superior Colliculus (SC) no cérebro humano. Figura retirada de [34].

O SC é estruturado em camadas, com um número de camadas que variam entre as espécies. As camadas superficiais são sensório-relacionadas e recebem informações oriundas dos olhos e de outros sistemas sensoriais. As camadas intermediárias são sensório-motoras e recebem informações visuais diretamente das camadas superficiais. Estas estruturas intermediárias são responsáveis por avaliar as informações sensoriais visuais e por representar, de forma retinotópica, as informações visuais do alvo sacádico. A partir das informações das camadas intermediárias, as camadas profundas, que também são sensório-motoras, são capazes de ativar os movimentos dos olhos e da cabeça.

2.5 Movimentos Oculares Sacádicos

Apenas uma fração da informação disponível da cena visual que recai na retina de nossos olhos pode ser processada em um dado momento. Esta fração é representada principalmente pelas fóveas [34] que, por serem as regiões da retina que possuem maior quantidade de fotorreceptores, possuem uma maior representação no córtex visual primário. Devido a estas limitações na acuidade visual da retina, os movimentos oculares são necessários para o processamento do campo visual.

Os movimentos sacádicos são os mais rápidos movimentos oculares. Sua principal função é direcionar o eixo visual para o local do objeto de interesse (mantendo-o alinhado com a fóvea), permitindo um grande nível de detalhamento da região alvo. Normalmente, o sistema visual executa várias sacadas por segundo e suas direções são selecionadas por um processo cognitivo no cérebro.

Na realização dos movimentos sacádicos, os neurônios do Superior Colliculus ativam e o cume do pico de ativação neural no mapa motor do SC representa a posição alvo da sacada [39], [40]. Enquanto a sacada está em andamento, um pico de atividade no SC se move gradualmente, indicando a diferença de posição do sistema óculo-motor [41] em relação ao objeto de interesse. Este pico de ativação caminha em direção à uma posição de referência no mapa motor que indica o fim do movimento de sacada. A Figura 7 ilustra esta situação.

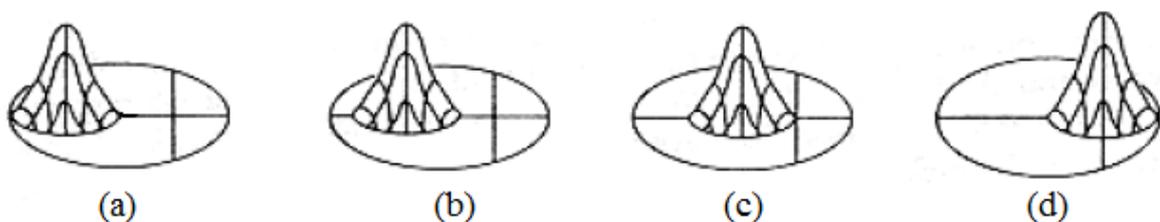


Figura 7: Mudança do padrão de atividade neural do Superior Colliculus durante a realização de movimentos sacádicos. Retirado de [42].

Na Figura 7 a intensidade da atividade neural é representada como a amplitude de um pico que se projeta a partir de uma superfície oval que representa uma região

do SC. A posição zero do mapa motor é representada pelo ponto de interseção das duas retas localizadas na superfície oval. A sequência (a), (b), (c), (d) mostra a movimentação do pico de ativação neural à medida que a sacada progride até atingir seu alvo, o que neste caso equivale ao pico de ativação residir sobre a posição zero do mapa motor. Em (a) temos a configuração do padrão de ativação neural do SC antes do início da sacada. Em (b) e (c) configurações intermediárias do padrão de ativação do SC durante a realização da sacada. Em (d) configuração do SC ao final da sacada.

A decomposição de uma olhada em movimentos de olho e de cabeça, bem como a trajetória exata dos movimentos oculares durante uma sacada, dependem da integração de sinais culliculares e não-culliculares por regiões do cérebro do sistema óculo-motor [41] ainda pouco conhecidas. Embora o SC codifique em seu padrão de ativação o alvo de uma olhada, ainda não está claro como este especifica os movimentos necessários para se chegar lá [43], [44].

Normalmente, as sacadas são pequenas e de maior duração em cenas complexas (como exemplo, cenas com objetos distratores semelhantes ao objeto de interesse), pois a pesquisa torna-se mais refinada do que em cenas simples. A Figura 8 ilustra a busca visual em duas situações: a) com objetos distratores semelhantes ao objeto de interesse e b) com objetos distratores diferentes do objeto de interesse. Em a) foi questionado ao observador que localizasse a letra “L” invertida e em b) o observador deveria encontrar um jarro contendo um cacto. Os pontos focais das sacadas são apresentados como círculos nas imagens.

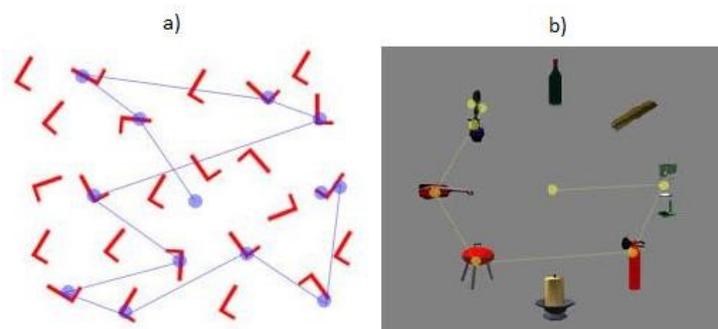


Figura 8: Comparação dos movimentos sacádicos entre cenas. Em a) observamos uma cena com distratores semelhantes ao objeto de interesse e em b) observamos uma cena com distratores distintos do objeto de interesse. Fonte: [45].

3 MODELO MATEMÁTICO-COMPUTACIONAL DE BUSCA VISUAL

Neste capítulo apresentamos um modelo matemático-computacional do sistema de busca visual biologicamente inspirado. Como apresentado anteriormente, a busca visual é o mecanismo por meio do qual, a partir do conhecimento prévio da imagem de um objeto de interesse, conseguimos encontrá-lo no campo visual se o mesmo nele estiver presente. A região cerebral responsável pela realização da busca visual, realizada através dos movimentos sacádicos dos olhos, é conhecida como Superior Colliculus.

Para criarmos um sistema computacional de busca visual a partir de um conjunto de imagens do mundo externo que busque similaridade com o sistema biológico [34] [46] modelamos o mapeamento retinotópico (as transformações na informação visual que ocorrem durante o caminho que ela faz a partir da retina até o córtex visual primário [47] [48]) e o processamento ocorrido na região do Superior Colliculus, envolvido diretamente no controle dos movimentos de sacada dos olhos. Tais modelos são apresentados a seguir.

3.1 Modelo do Mapeamento Retinotópico

Uma representação de uma imagem de círculos concêntricos no córtex visual primário foi obtida pelo pesquisador Tootell [38] através da aplicação de uma substância radioativa (2-deoxyglucose) em um dos olhos de um primata e do controle da fixação da fóvea [47] deste olho no centro da imagem. Após um determinado tempo, o primata foi sacrificado e a parte correspondente ao córtex visual primário do hemisfério esquerdo de seu cérebro foi recortada, aplainada e posicionada sobre um filme fotográfico sensível à radiação que, quando revelado, mostrou a imagem apresentada na Figura 9. Podemos observar, nesta figura, que os círculos concêntricos na imagem se tornam quase retas no córtex visual primário e

as regiões circunscritas pelos círculos mais internos, de menor área na imagem, ocupam uma área muito maior em sua representação, indicando a alta acuidade visual destas regiões.

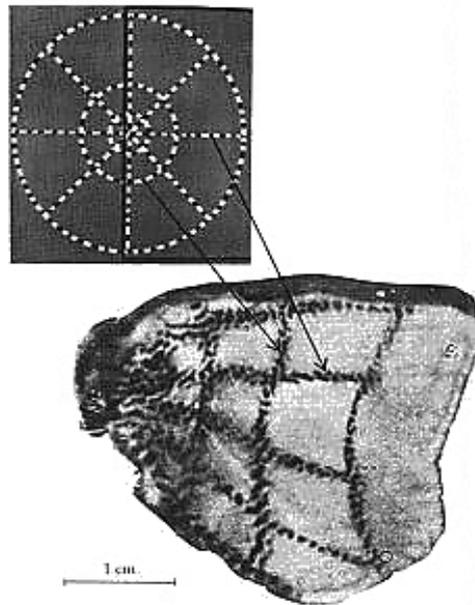


Figura 9: Representação de uma imagem de círculos concêntricos projetada na retina em V1. Figura retirada de [38].

Podemos modelar matematicamente o mapeamento que ocorre nas informações visuais entre a retina e o córtex visual primário através de uma transformação log-polar. Esta transformação realiza o mapeamento dos pontos do plano cartesiano (coordenadas x e y) para pontos no plano log-polar (coordenadas ρ e θ) de acordo com Equação 1 e Equação 2.

$$R = \sqrt{(x - x_c)^2 + (y - y_c)^2} \rightarrow \rho \propto \log(R) \quad \text{Equação 1}$$

$$\theta = \arctan\left(\frac{(y - y_c)}{(x - x_c)}\right) \rightarrow \phi \propto \theta \quad \text{Equação 2}$$

A transformação log-polar de uma imagem com centro (x_c, y_c) correspondente ao centro de atenção no campo visual é apresentada na Figura 10. Podemos perceber que o círculo vermelho na imagem à esquerda torna-se uma linha em sua representação à direita.

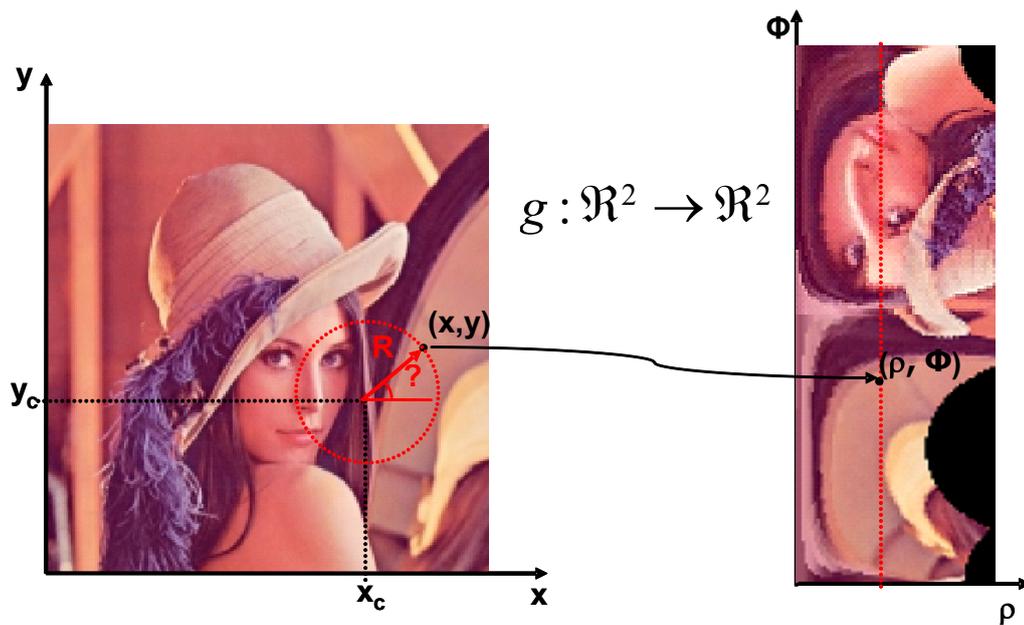


Figura 10: Transformada log-polar. Figura retirada de [42].

Uma característica da transformação log-polar é que, devido ao comportamento logarítmico da variável ρ , que é diretamente proporcional ao logaritmo da distância R da origem ao ponto (x_c, y_c) , as regiões mais próximas ao centro das coordenadas cartesianas possuem uma maior representação (uma melhor definição no plano log-polar) e as regiões mais afastadas do centro possuem uma menor representação (uma pior definição no plano log-polar). Tal característica é semelhante ao que ocorre no sistema visual humano, em que a região da fóvea possui uma alta resolução em relação a regiões periféricas. Este comportamento pode ser observado na Figura 11. Nesta figura apresentamos um exemplo da aplicação da transformação log-polar numa imagem seguido do mapeamento log-polar inverso. Uma imagem cartesiana bidimensional (Figura 11a) de 128×128 pixels sofre uma transformação log-polar com uma amostragem de 64×32 (ρ e θ respectivamente) representada na (Figura 11b). Realizando a transformação log-polar inversa (Figura 11c) é possível verificar a perda de definição à medida em que se afasta do centro da imagem.

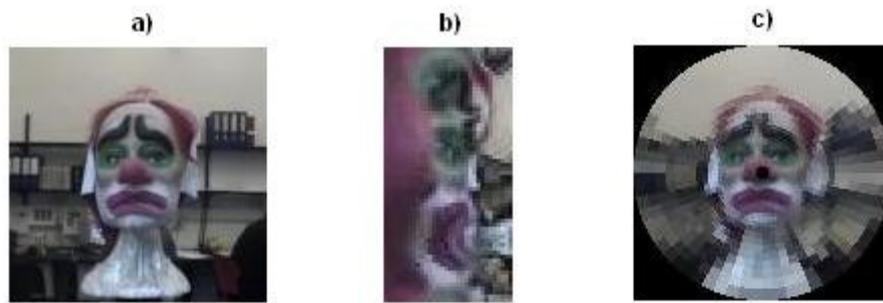


Figura 11: Aplicação da transformação log-polar seguida pelo mapeamento inverso. Figura retirada de <http://omni.isr.ist.utl.pt/~alex/Projects/TemplateTracking/logpolar.htm>.

Não foi empregado neste trabalho a transformada log-polar da mesma forma como apresentada, mas sim uma variante que emula mais precisamente o mapeamento retina-córtex visual primário [49]. Um resultado da aplicação desta variante pode ser visto na Figura 12, onde vizinhanças na imagem ao redor do ponto de fixação, ou seja, a fóvea do modelo proposto (o centro do círculo), são respeitadas na projeção log-polar (retinotopia), como ocorre no cérebro (por meio do corpo caloso [50]). Isso não ocorre na transformada da Figura 10.

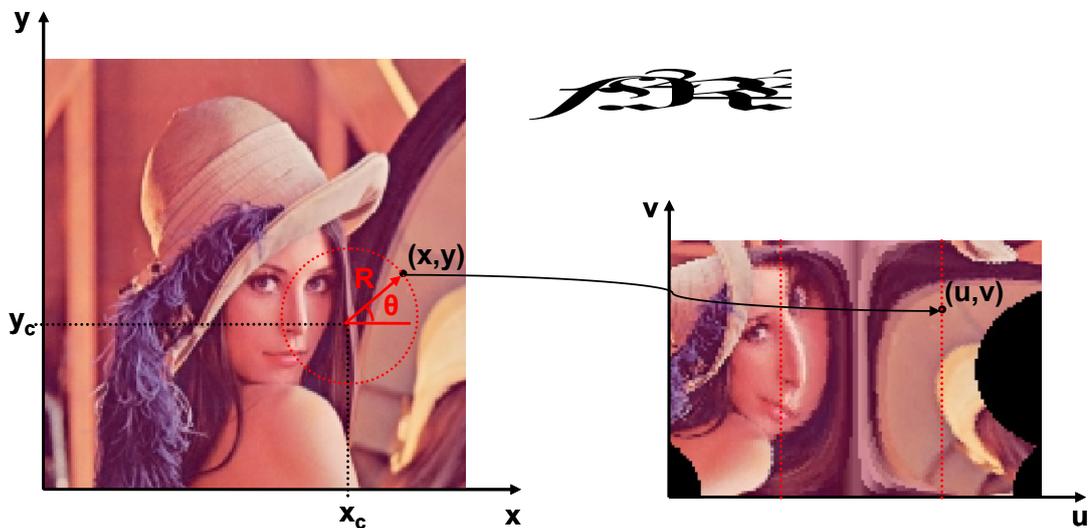


Figura 12: Transformada log-polar empregada pelo Grupo de Pesquisa em Cognição Visual do LCAD. Figura retirada de [42].

3.2 Redes Neurais Sem Peso VG-RAM

Redes Neurais Sem Peso (RNSP) são ferramentas de reconhecimento de padrões baseadas em *Random Access Memories* (RAM) que não armazenam conhecimento em suas conexões, mas em memórias do tipo RAM dentro dos neurônios (nodos da rede) [51].

As sinapses de cada neurônio coletam um vetor de *bits* da entrada da rede que é usado como o endereço da RAM e o valor armazenado neste endereço é a saída do neurônio. O treinamento pode ser feito em um único passo e consiste em armazenar a saída desejada no endereço associado com o vetor de entrada do neurônio.

As RNSP do tipo *Virtual Generalizing RAM* (VG-RAM) [23] são redes neurais baseadas em RAM que somente requerem capacidade de memória para armazenar os dados relacionados ao conjunto de treinamento. Os neurônios VG-RAM armazenam os pares entrada-saída observados durante o treinamento, ao invés de apenas a saída. Na fase de teste, as memórias dos neurônios VG-RAM são pesquisadas mediante a comparação entre a entrada apresentada à rede e todas as entradas dos pares entrada-saída aprendidos. A saída de cada neurônio VG-RAM é determinada pela saída do par cuja entrada é a mais próxima da entrada apresentada.

As RNSP do tipo VG-RAM são baseadas em tabelas-verdade (*lookup tables*) [23,52]. A Tabela 1 ilustra a tabela-verdade de um neurônio VG-RAM com três sinapses (w_1 , w_2 e w_3). Esta tabela-verdade contém três pares entrada-saída (*entrada#1*, *entrada#2* e *entrada#3*) que foram armazenadas durante a fase de treinamento. Durante a fase de teste, quando um vetor de entrada é apresentado à rede, o algoritmo de teste VG-RAM calcula a distância entre este vetor de entrada e cada entrada dos pares entrada-saída armazenados na tabela-verdade. A função de distância adotada pelos neurônios VG-RAM é a distância de *Hamming*, ou seja, o número de bits diferentes entre dois vetores de bits de igual tamanho. Se existir mais

do que um par na mesma distância mínima da entrada apresentada, a saída do neurônio é escolhida aleatoriamente entre esses pares.

Tabela 1: Tabela-verdade de um neurônio da RNSP VG-RAM. Retirada de [53].

<i>Tabela verdade</i>	w_1	w_2	w_3	N
<i>entrada #1</i>	1	1	0	<i>saída 1</i>
<i>entrada #2</i>	0	0	1	<i>saída 2</i>
<i>entrada #3</i>	0	1	0	<i>saída 3</i>
	↑	↑	↑	↓
<i>nova entrada</i>	1	0	1	<i>saída 2</i>

No exemplo da Tabela 1 a distância de *Hamming* entre o vetor da nova entrada e a *entrada#1* é dois, porque ambos os bits w_2 e w_3 não são iguais aos bits w_2 e w_3 do vetor da nova entrada. A distância da *entrada#2* é um, porque w_1 é o único bit diferente. Já a distância da *entrada#3* é três, dado que nenhum bit é igual. Portanto, para este vetor da nova entrada, o algoritmo avalia a saída do neurônio, N , como “*saída 2*”, pois é o valor de saída armazenado na *entrada#2* que possui a menor distância de *Hamming*.

As RNSP VG-RAM já foram empregadas com sucesso por pesquisadores do LCAD (<http://www.lcad.inf.ufes.br>) em diversos problemas relacionados à Ciência da Cognição e Aprendizado de Máquina, como controle de vergência em sistemas de visão artificial [54], reconhecimento de faces [55], categorização automática de texto [56,57,58,59] e reconhecimento de placas de trânsito [28] [29]. Estes exemplos têm em comum o modelo de memória distribuída (individual) dos neurônios. Entretanto, o sistema de busca visual descrito neste trabalho emprega o modelo de memória compartilhada (ou coletiva) dos neurônios.

O modelo de memória compartilhada implica em um maior gasto de tempo para computar por completo a saída da rede neural. No modelo de memória distribuída tradicional, este tempo é diretamente proporcional ao número de entradas

armazenadas durante a fase de treinamento, enquanto que com memória compartilhada, este passa a ser proporcional ao número de entradas de treinamento vezes o número de neurônios, uma vez que a memória é consultada de forma coletiva.

3.2.1 Redes Neurais Sem Peso VG-RAM Fat-Fast

Uma otimização das RNSP VG-RAM são as RNSP VG-RAM *Fat-Fast* [60], que empregam a estrutura de dados *hash* multi-indexada para proporcionar uma busca rápida na memória dos neurônios VG-RAM. Esta técnica aumenta o consumo de memória (*fat*) e reduz o tempo de busca (*fast*), daí o nome *fat-fast*. Nesta abordagem, várias tabelas *hashs* são criadas como memória de cada neurônio VG-RAM considerando o espaço binário definido pelo vetor de entrada I .

A Figura 13 apresenta uma visão geral da arquitetura da memória dos neurônios *Fat-Fast*. Nesta figura, um neurônio *Fat-Fast* X lê, via suas sinapses, o vetor binário de entrada I . A figura ilustra o processo de mapeamento de um vetor de entrada, I , de tamanho igual a p bits, em índices, $B_{k,v}$, que armazenam as referências para o conjunto de linhas da memória que devem ser inspecionadas durante a busca na memória, ilustradas por setas. O vetor de entrada I é inicialmente dividido em sub-vetores, $\mathbf{V} = \{V_1, \dots, V_k \dots, V_h\}$, onde $V_k = \{0, 1, \dots, v, \dots, 2^{p/h}\}$. Estes sub-vetores são utilizados para endereçar a tabela *hash* multi-indexada, $\mathbf{H} = \{H_1, \dots, H_k \dots, H_h\}$. Cada endereço, V_k , aponta para um índice, $B_{k,v}$, de linhas, L , que devem ser consideradas na pesquisa. Note que esta ilustração utiliza códigos binários, V_k , de 3 bits apenas para maior clareza.

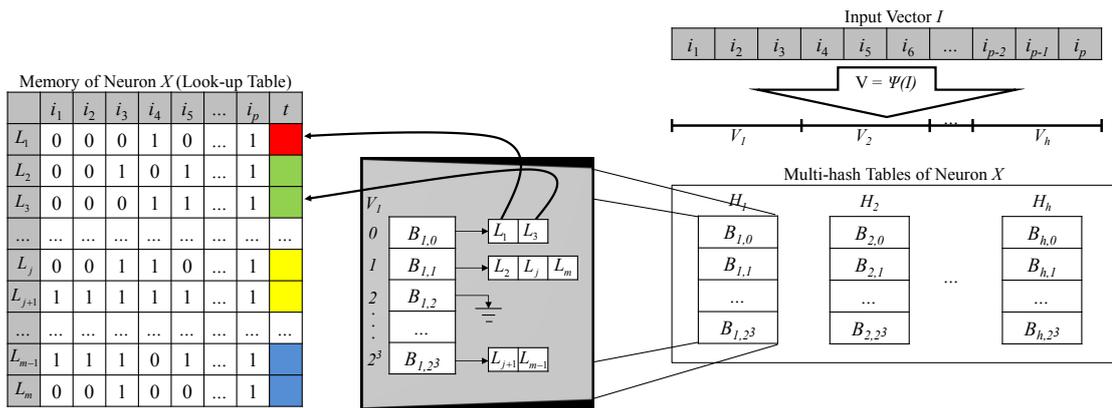


Figura 13: Visão geral da arquitetura dos neurônios VG-RAM Fat-Fast. Figura retirada de [60].

Na fase de treino de um neurônio *Fat-Fast*, as memórias dos neurônios são preenchidas com pares entrada-saída seguindo o mesmo procedimento dos neurônios VG-RAM. Além disso, é necessário preencher a tabela *hash* multi-indexada, H , com as referências de cada par aprendido. Para isso, o código binário V é extraído a partir de I_j usando $\Psi(I_j)$ e uma referência para L_j é adicionada ao respectivo índice de cada tabela *hash* H .

Na fase de teste, inicialmente, utiliza-se a função $\Psi(I)$ para recuperar V . Em seguida, cada V_k é utilizado como endereço para o índice $B_{k,v}$. Por fim, o vetor de entrada I_j de cada linha da memória $L_j = (I_j, t_j)$ de cada índice selecionado é examinado como candidato a entrada mais próxima ao vetor binário de entrada I . Note que todos os candidatos são comparados com I utilizando a distância de *Hamming* entre vetores de *bits* completos, ou seja, os p bits de I e I_j . A saída do neurônio VG-RAM *Fat-Fast* é o rótulo t_j da linha mais próxima L_j . Se existir mais do que uma linha candidata na mesma distância mínima da entrada apresentada, a saída do neurônio é escolhida aleatoriamente entre as candidatas.

Caso existam muitas linhas em um ou mais dos índices selecionados, o tempo de busca pode aumentar substancialmente. Assim, neste caso, um máximo de r candidatos em cada índice é selecionado aleatoriamente para comparação com o vetor de entrada I , onde r é um parâmetro da rede. É importante ressaltar, ainda, que utiliza-se o modelo de memória compartilhada, implicando em um maior gasto

de tempo para computar por completo a saída da rede neural, uma vez que a memória é consultada de forma coletiva.

3.3 Modelo do Superior Colliculus

Apresentamos na Figura 14 a arquitetura de RNSP VG-RAM proposta para modelar o Superior Colliculus, que utiliza o modelo de memória compartilhada com uma única camada neural composta por neurônios VG-RAM (que podem ser neurônios VG-RAM ou VG-RAM *Fat-Fast*) que estão conectados a uma camada de entrada por um conjunto de sinapses. Cada neurônio apresenta um valor de ativação de saída (círculos coloridos) obtidos na tabela-verdade (memória compartilhada). Neste caso, cada neurônio deve calcular a distância de *Hamming* entre a sua entrada e as entradas armazenadas por todos os neurônios da rede. A saída será a associada à entrada da tabela-verdade que possuir a menor distância de *Hamming*, podendo esta ter sido memorizada por qualquer neurônio da rede.

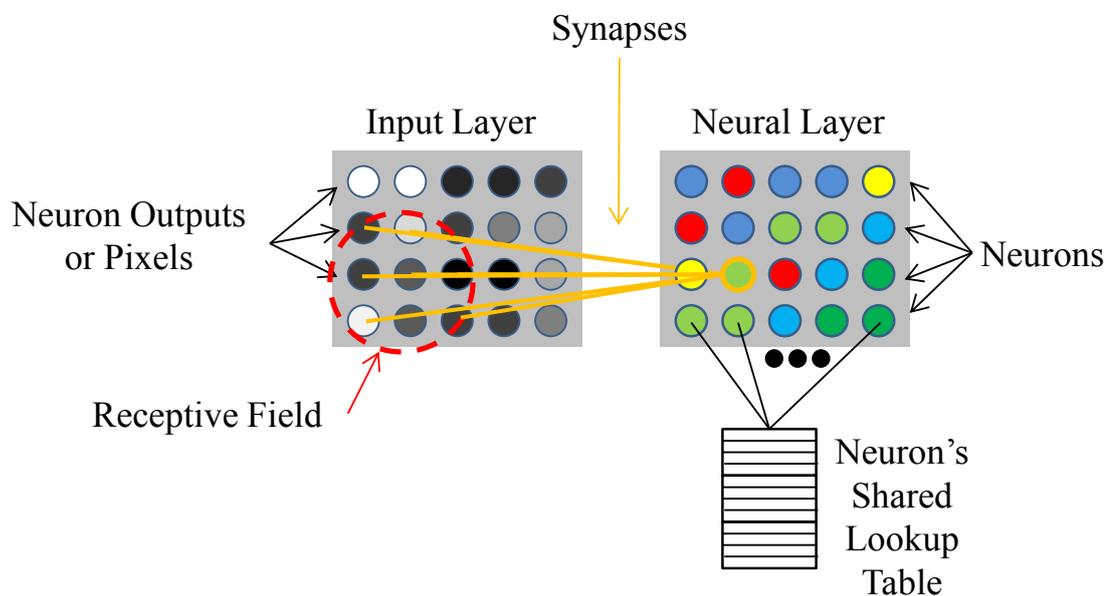


Figura 14: Arquitetura proposta do Superior Colliculus (SC). Figura retirada de [61].

No modelo proposto, o centro do campo receptivo dos neurônios é mapeado para a camada de entrada de acordo com uma função log-polar inversa que emula a

fóvea no sistema, uma vez que aumenta o número de campos receptivos de neurônios ao longo da região central da camada de entrada (Figura 15). A camada de entrada, que representa a retina no sistema, recebe uma porção de uma imagem de um *frame* de vídeo. O centro desta região é o centro de atenção do sistema. Os neurônios da camada neural ativam (saída com cor diferente de preto) quando o objeto de interesse está em seu campo visual.

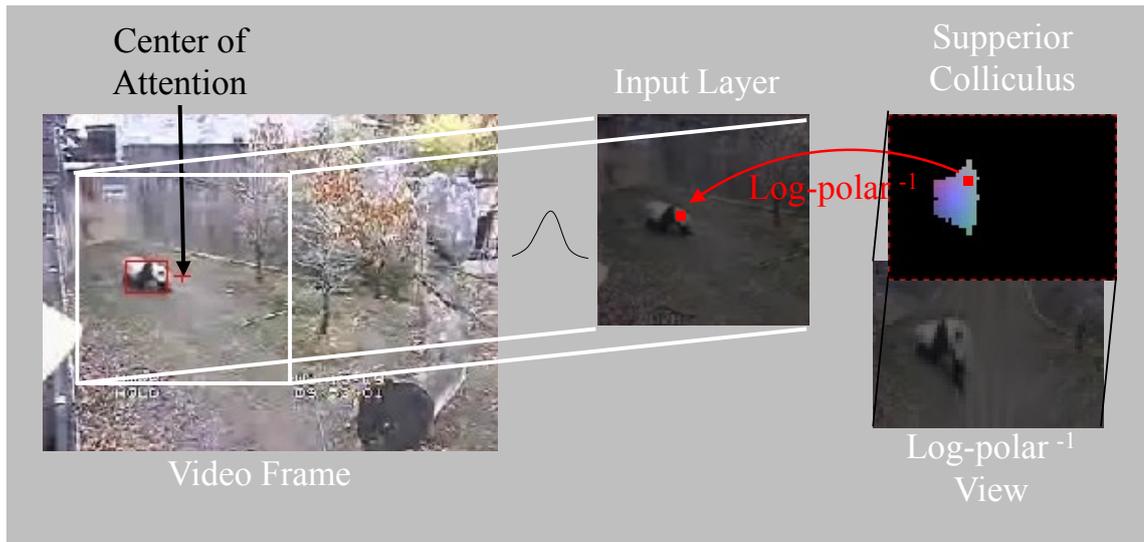


Figura 15: O modelo do Superior Colliculus (SC) proposto. Figura retirada de [61].

Antes de chegar à entrada da rede neural a imagem I é recortada (considerando o tamanho da camada de entrada e o centro de atenção) e sofre um ajuste de escala e uma suavização gaussiana, de modo que a imagem na camada de entrada da rede seja o mais semelhante à incidente no olho humano (aproximação do objeto de interesse e ajuste de foco ocular) [50]. O processo de ajuste de escala será discutido posteriormente.

Sinapses de RNSP coletam apenas um *bit* (0 ou 1) de entrada. Para permitir o uso de entradas não binárias foram utilizadas *minchinton cells* [62]. Na arquitetura neural empregada, cada sinapse, w_t , forma uma *minchinton cell* com a próxima, w_{t+1} (a última sinapse forma uma *minchinton cell* com a primeira). Cada uma destas *minchinton cells* retorna 1 se a sinapse w_t está conectada a um elemento da camada de entrada cujo valor é maior que o valor do elemento ao qual a sinapse w_{t+1} está conectada; caso contrário a *minchinton cells* irá retornar 0. Existem várias formas de

comparar os elementos da entrada. Neste trabalho, essa comparação é feita por meio da sinapse de cor seletiva composta [42].

As sinapses são conectadas à entrada bidimensional da rede, Φ , composta de $u \times v$ pixels, segundo um padrão de interconexão sináptico que segue uma distribuição aleatória bidimensional Normal com variância σ^2 centrada no pixel φ_{μ_k, μ_l} , onde as coordenadas μ_k e μ_l de Φ são determinados pela transformação log-polar inversa das coordenadas i e j do neurônio n_{ij} da camada neural N , ou seja, a distribuição das coordenadas k e l dos pixels de Φ aos quais n_{ij} se conectam via sinapses seguem as PDF (*Probabilistic Density Function*):

$$\omega_{\mu_k, \sigma^2}(k) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(k-\mu_k)^2}{2\sigma^2}}$$

$$\omega_{\mu_l, \sigma^2}(l) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(l-\mu_l)^2}{2\sigma^2}}$$

onde σ é parâmetro da arquitetura e as coordenadas μ_k e μ_l são calculadas por:

$$\mu_k = \frac{u}{2} + d \cdot \cos(\theta)$$

$$\mu_l = \frac{v}{2} + d \cdot \sin(\theta)$$

onde

$$d = \frac{u}{2} \cdot \left(\frac{\alpha^{\left| \frac{i-m/2}{m/2} \right|} - 1}{\alpha - 1} \right)$$

$$\theta = \begin{cases} \pi \cdot \left(\frac{3n}{2} - \frac{j}{n} \right) + \frac{\pi}{2n}; & \text{se } k < \frac{m}{2} \\ \pi \cdot \left(\frac{3n}{2} + \frac{j}{n} \right) + \frac{\pi}{2n}; & \text{se } k > \frac{m}{2} \end{cases}$$

e α é o log-factor da função log-polar e é parâmetro da arquitetura.

Este padrão de interconexão sináptica é comum em neurônios biológicos [34]. Entretanto, uma diferença do padrão de interconexão sináptica adotado na aplicação de busca visual em relação ao empregado em outras aplicações com RNSP VG-RAM é que, embora aleatório, é o mesmo para todos os neurônios.

Durante o treino da camada neural, inicialmente, um frame de vídeo é dado e o centro de atenção do sistema é movido para o centro da caixa delimitadora do objeto de interesse. Em seguida, a camada neural é pintada com a cor que representa a posição do centro do campo receptivo do neurônio em relação ao centro da caixa delimitadora do objeto (ver Figura 16). Por fim, os neurônios são treinados para responder preto, se o centro de seu campo receptivo está fora da caixa delimitadora do objeto, ou a cor específica que representa o seu deslocamento em relação ao centro da caixa delimitadora do objeto. Portanto, os neurônios que aprendem a cor preto aprendem o fundo da imagem e os neurônios que aprendem qualquer outra cor diferente de preto aprendem sobre a imagem do objeto.

O padrão de cor utilizado para treinar a camada neural é uma imagem RGB com um significado especial para cada um dos três canais de cor: canal vermelho (r_c), canal verde (g_c) e canal azul (b_c) (ver Figura 16). O canal vermelho (r_c) vai de 0 a 254 e representa uma posição no eixo vertical da imagem do objeto de interesse, em que o valor 0 indica a parte mais baixa do objeto, o valor 127 indica o centro vertical do objeto e o valor 254 indica a parte mais alta do objeto. O valor 255 não é usado para permitir um intervalo com um número ímpar de elementos, possibilitando um único centro. O canal verde (g_c) vai de 0 a 254 e representa uma posição no eixo horizontal da imagem do objeto de interesse, em que o valor de 0 indica a parte mais a esquerda do objeto, o valor de 127 indica que o centro horizontal do objeto e o valor de 254 indica a parte mais à direita do objeto. O canal azul (b_c) vai de 0 a 255 e representa o inverso da distância a partir do centro do objeto de interesse, em todos os sentidos, em que o valor 255 indica o centro do objeto de interesse e o valor 0 indica a parte mais afastada do centro deste objeto.

Esta organização de cor permite a associação da saída de cada neurônio, $n_{i,j}$, com um vetor de deslocamento específico, $V_{i,j} = (x, y)$, que representa o

deslocamento entre o centro de atenção do sistema e o centro do objeto de interesse. Este vetor pode ser calculado em função dos canais verde e vermelho da saída do neurônio e das coordenadas do neurônio na camada neural, como representado na Equação 3:

$$V_{i,j} = F(r_c, g_c, i, j).$$

Equação 3

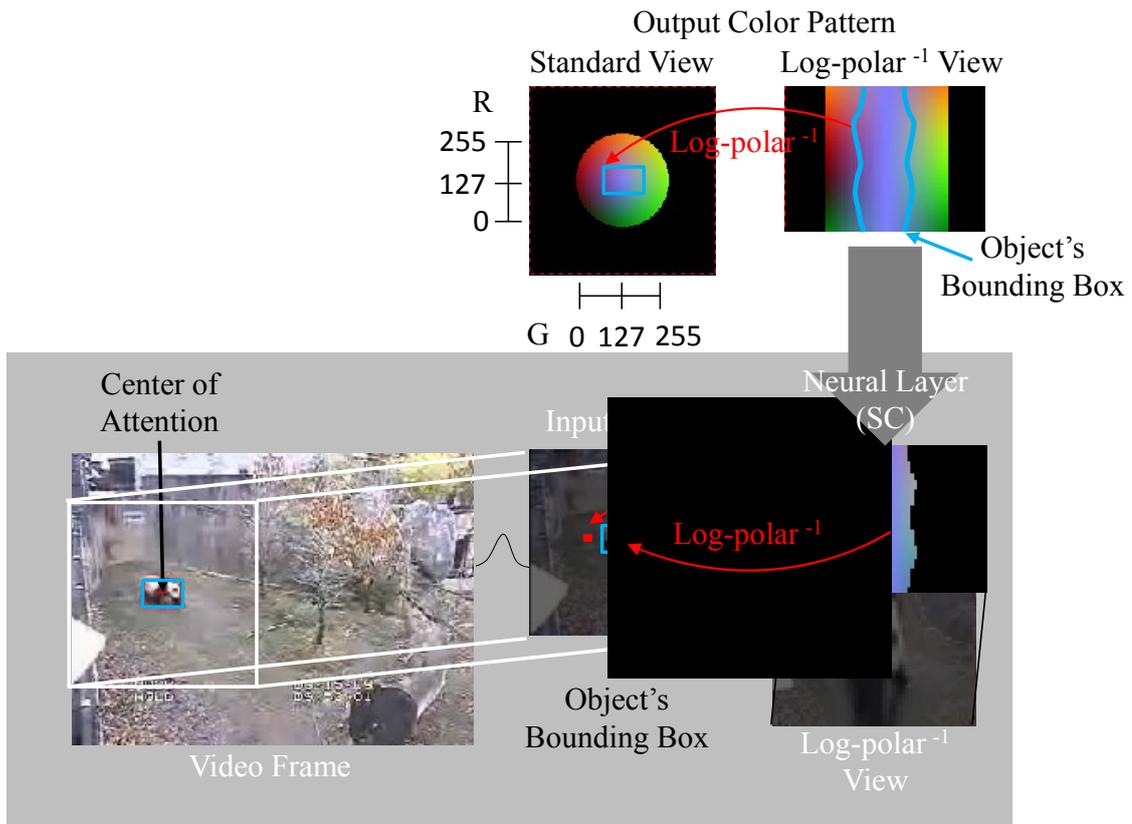


Figura 16: Ilustração da fase de treino. O centro de atenção é inicialmente movido para o centro do objeto a ser aprendido. Em seguida, a camada neural é pintada com a cor de saída esperada por cada neurônio. Neurônios presentes na caixa delimitadora do objeto são considerados ativos e têm a cor diferente de preto associada a eles. Neurônios fora da caixa delimitadora do objeto são considerados não-ativos e têm a cor preta associada a eles. As cores são organizadas de forma que a cor da saída de um neurônio diz qual é o deslocamento deste neurônio do centro do objeto de interesse. Figura retirada de [61].

Durante a fase de teste, um *frame* de vídeo é dado e o centro de atenção do sistema é inicialmente movido para a região a ser pesquisada. Em seguida, os neurônios respondem de acordo com o vetor binário de entrada amostrado em seus campos receptivos. Os neurônios respondem preto quando o padrão observado em seus campos receptivos é semelhante às regiões anteriormente aprendidas como

plano de fundo ou respondem uma cor diferente de preto quando o padrão observado em seus campos receptivos é semelhante às regiões anteriormente aprendidas como objeto de interesse. Neurônios ativados oferecem informações sobre a localização precisa do centro do objeto de interesse codificada a partir dos valores dos canais vermelho (r_c) e verde (g_c). A Figura 15 mostra um exemplo da fase de teste com o centro de atenção um pouco distante do centro do objeto de interesse (urso panda) e a Figura 17 mostra um exemplo da fase de teste com o centro de atenção perto do centro do objeto de interesse (urso panda).

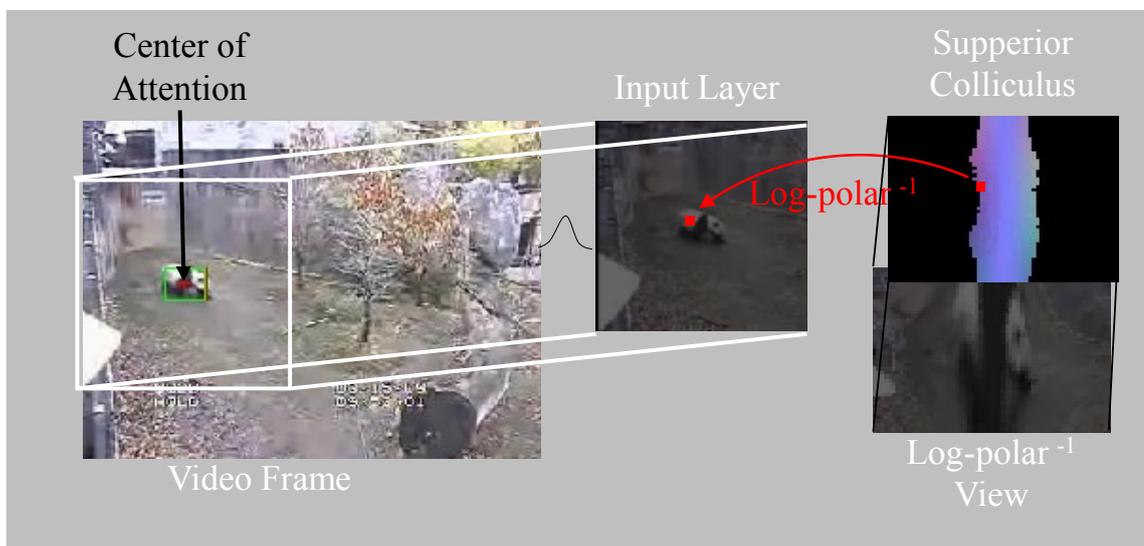


Figura 17: Ilustração da fase de teste com o centro de atenção perto do centro do objeto de interesse (panda). Figura retirada de [61].

3.4 O Movimento Sacádico dos Olhos

A fim de determinar o centro de um objeto de interesse previamente aprendido, cada neurônio ativado n_{ij} contribui com um voto para a provável localização do centro deste objeto através do vetor V_{ij} e cada voto é ponderado por uma medida de confiança w_{ij} associada a cada neurônio n_{ij} (ver detalhes a seguir). Os votos são acumulados em uma matriz acumuladora inicialmente zerada que possui as mesmas dimensões da camada de entrada e representa todas as possíveis localizações espaciais que o centro do objeto de interesse pode ter, caso este esteja presente na

imagem da camada de entrada. A coordenada da célula da matriz acumuladora mais votada pelos neurônios é selecionada como sendo a coordenada alvo do movimento sacádico e a pontuação acumulada é utilizada para medir a confiança do movimento. Na Figura 18 apresentamos uma visão geral do processo de determinar o alvo do movimento sacádico. Nos parágrafos seguintes, cada etapa deste processo é descrita em detalhes.

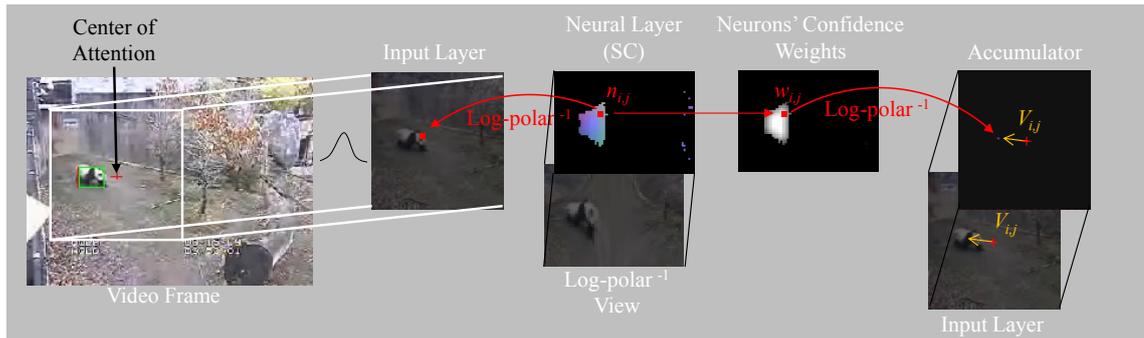


Figura 18: Ilustração da arquitetura usada para determinar o alvo do movimento sacádico. Cada neurônio ativado, $n_{i,j}$, da camada neural contribui com um voto para a provável localização, $V_{i,j}$, do centro do objeto de interesse. Cada voto é ponderado por uma medida de confiança da resposta de um neurônio, $w_{i,j}$. Os votos são armazenados em uma matriz acumuladora, onde as células representam as possíveis localizações espaciais do objeto de interesse. O local mais votado é escolhido como alvo do movimento sacádico. Figura retirada de [61].

Para contribuir com um voto para o provável local do centro do objeto de interesse, cada neurônio tem que decidir sobre uma localização na camada de entrada para votar. Esta localização é representada pelo vetor deslocamento $V_{i,j}$ a partir do atual centro de atenção (no centro da camada de entrada). O deslocamento $V_{i,j}$ de cada neurônio ativo $n_{i,j}$ é calculado a partir da localização real do neurônio na camada neural e da saída do neurônio, ou seja, do deslocamento aprendido do centro do objeto de interesse representado pela cor da saída do neurônio. Mais precisamente, $V_{i,j}$ é dado por:

$$V_{i,j} = F(r_c, g_c, i, j) = L_{i,j}(r, \theta) - C_{i,j}(r_c, g_c)$$

onde $L_{i,j}(r, \theta)$ é o vetor que representa a localização do centro do campo receptivo do neurônio $n_{i,j}$ na camada de entrada em relação ao centro de atenção e $C_{i,j}(r_c, g_c)$ é o

vetor que representa o deslocamento do centro do objeto de interesse para o centro do campo receptivo do neurônio $n_{i,j}$ na camada de entrada (ver Figura 19).

O vetor $L_{i,j}(r, \theta)$ é, de fato, o mapeamento log-polar inverso das coordenadas (i, j) do neurônio $n_{i,j}$ da camada neural N conforme discutido anteriormente.

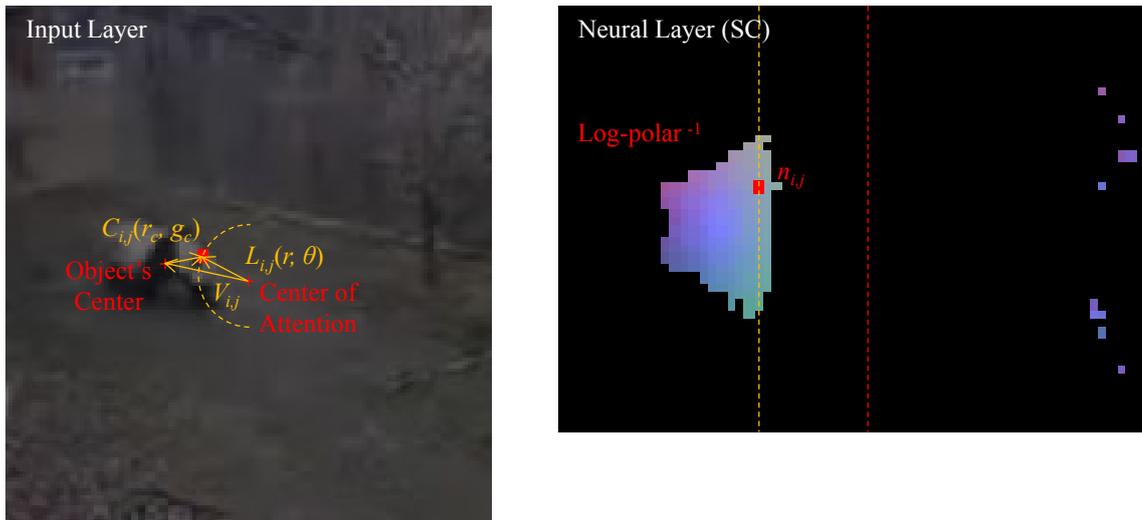


Figura 19: O deslocamento de um neurônio. O deslocamento $V_{i,j}$ é obtido adicionando o vetor $L_{i,j}(r, \theta)$ ao vetor $-C_{i,j}(r_c, g_c)$. $L_{i,j}(r, \theta)$ corresponde à localização do centro do campo receptivo do neurônio $n_{i,j}$ na camada de entrada. $C_{i,j}(r_c, g_c)$ corresponde ao deslocamento do centro do objeto de interesse em relação ao centro do campo receptivo do neurônio $n_{i,j}$. Figura retirada de [61].

O vetor $C_{i,j}(r_c, g_c)$ é o deslocamento que foi aprendido durante a fase de treino, o qual pode ser calculado usando a cor da saída do neurônio $n_{i,j}$, como demonstrado na Equação 4. Caso o centro de atenção coincida com o centro do objeto de interesse $L_{i,j}(r, \theta) = C_{i,j}(r_c, g_c)$ e $V_{i,j} = (0, 0)$.

$$C(r_c, g_c) = (r_c - 127, g_c - 127).$$

Equação 4

Os pesos dos votos dos neurônios $w_{i,j}$ são calculados considerando-se a proximidade de dois neurônios ativos em N e a proximidade de seus alvos em Φ . Baseamos esta hipótese no fato de que, se dois neurônios vizinhos, $n_{i,j}$ e $n_{i,j+1}$, têm cores de ativação corretas, ambos devem sinalizar as mesmas coordenadas do centro do objeto de interesse, ou seja, $V_{i,j} = V_{i,j+1}$. Expandindo esta hipótese a todos os vizinhos do neurônio $n_{i,j}$ e considerando uma janela de vizinhança de raio s (a janela de tamanho $2s+1 \times 2s+1$ pixels), o peso $w_{i,j}$ é dado por:

$$w_{i,j} = e^{-T_{i,j}/(2*\beta^2)}, e$$

$$T_{i,j} = \sum_{a=-s}^s \sum_{b=-s}^s |V_{i,j} - V_{i+a,j+b}|,$$

onde β^2 é um parâmetro do modelo. Com essa função de peso, os neurônios que respondem de acordo com os seus vizinhos terão pesos mais elevados. Na nossa aplicação, s foi empiricamente escolhido para ser igual a 3 e β igual a 10.

Uma matriz acumuladora de votos é utilizada para determinar o alvo do movimento sacádico. O vetor deslocamento $V_{i,j}$ é calculado para todos os neurônios e os pesos $w_{i,j}$ são computados. A célula da matriz acumuladora correspondente a localização indicada pelo vetor $V_{i,j}$ de cada neurônio é incrementada com o peso $w_{i,j}$ correspondente (veja Figura 20). A célula mais votada da matriz acumuladora é selecionada como alvo do movimento sacádico e o deslocamento V do centro de atenção corrente é realizado. O valor acumulado na célula vencedora é utilizado como a confiança do movimento sacádico.

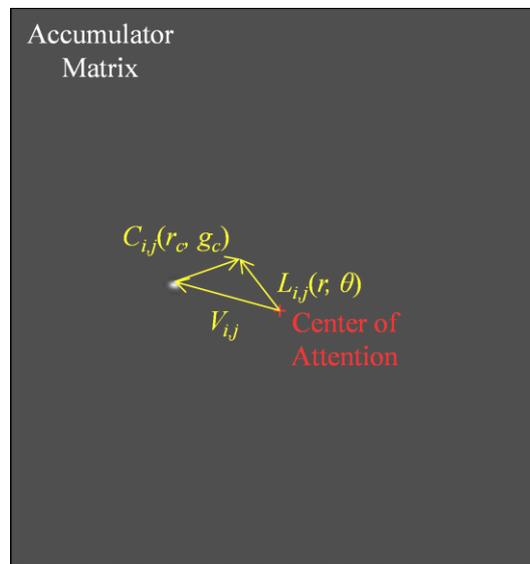


Figura 20: Ilustração da acumulação de votos para a possível localização espacial do objeto de interesse. Cada célula no acumulador corresponde à localização indicada pelo vetor de deslocamento $V_{ij} = L_{ij}(r, \theta) - C_{ij}(r_c, g_c)$ de cada neurônio n_{ij} . Figura retirada de [61].

4 RASTREAMENTO DE OBJETOS COM RNSP VG-RAM

Neste capítulo apresentamos o sistema de rastreamento visual biologicamente inspirado baseado em VG-RAM WNN. O sistema destina-se a rastrear, a longo prazo, um objeto em um vídeo. Resumidamente, uma caixa delimitadora em torno do objeto de interesse é definida no primeiro *frame* de um vídeo e o sistema de rastreamento visual determina a localização do objeto em *frames* subsequentes. Além disso, o sistema indica a presença ou não do objeto nos *frames*. A localização do objeto é dada por uma caixa delimitadora.

A Figura 21 apresenta um exemplo de entrada e saída do sistema de rastreamento visual. O objeto de interesse, usado para treinar o sistema no primeiro *frame*, é apresentado em (a) delimitado em vermelho. A saída do sistema em um *frame* aleatório subsequente (b) é mostrada em verde em conjunto com uma caixa delimitadora anotada manualmente (*ground-truth*) em vermelho. Um *frame* sem o objeto, situação na qual o sistema deve responder sem caixa delimitadora, é mostrado em (c).

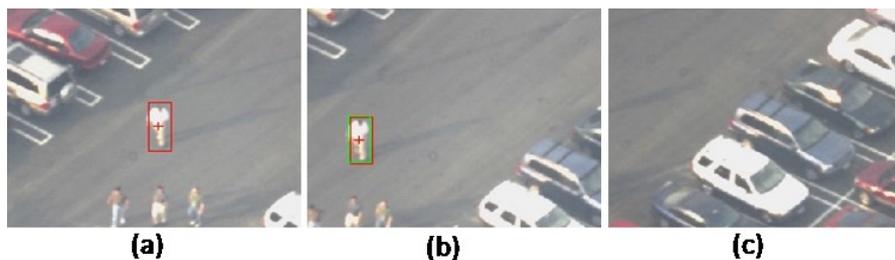


Figura 21: Resultados do sistema de rastreamento visual. Em (a), a caixa delimitadora do objeto de interesse é apresentada em vermelho. Em (b), a saída do sistema para um *frame* diferente é mostrada em verde em conjunto com uma caixa delimitadora anotada manualmente em vermelho. Em (c), o objeto não é visível e o sistema não apresenta nenhuma caixa delimitadora. Figura retirada de [61].

O sistema de rastreamento visual compreende quatro módulos: (Re)Learning, Tracking, Detection and Validation (Figura 22). O módulo (Re)Learning é responsável por treinar o sistema com o objeto manualmente anotado no primeiro *frame* e por retreinar com o objeto rastreado em *frames* subsequentes a fim de

reforçar a descrição atual do objeto aprendida pelo sistema. O módulo Tracking é responsável por rastrear o objeto, *frame a frame*, considerando a última localização do mesmo. O módulo Detection é responsável por detectar o objeto quando este retorna a cena, após oclusão. O módulo Validation é responsável por (i) decidir se é necessário o retreino do objeto no sistema, (ii) se o processo de rastreamento deve continuar ou (iii) se o processo de detecção deverá ser iniciado. A partir da decisão do módulo Validation, o sistema retorna as coordenadas da caixa delimitadora do objeto ou a informação de que o objeto não é visível na cena.

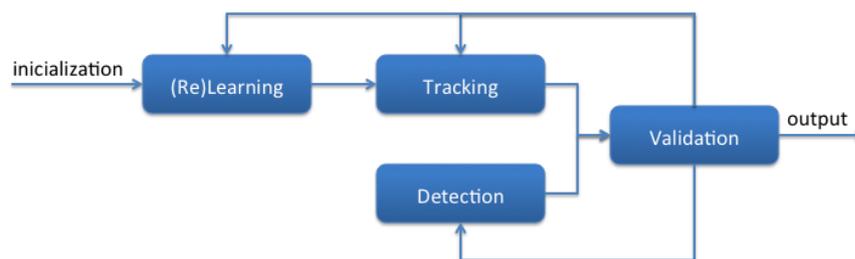


Figura 22: Diagrama de bloco do sistema de rastreamento visual. Figura retirada de [61].

Nas seções seguintes detalhamos cada módulo do sistema de rastreamento visual - (Re)Learning, Tracking, Detection e Validation – e apresentamos como estes interagem com o sistema de movimentos sacádicos implementado com VG-RAM WNN apresentado no capítulo anterior.

4.1 (Re)Learning

O módulo (Re)Learning é responsável por treinar a camada neural do modelo de Superior Colliculus proposto e pode ser dividido em duas etapas: treino (*learning*) e retreino (*relearning*). A etapa de treino compreende o treino inicial, na qual o sistema aprende o objeto manualmente anotado no primeiro *frame* do vídeo. A etapa de retreino é a etapa em que o sistema aprende novas aparências do objeto de interesse, adaptando-se a eventuais alterações de aparência ocorridas ao longo do tempo.

O treino inicial do sistema é realizado conforme apresentado no capítulo anterior, ou seja, o centro de atenção é movido para o centro do objeto de interesse e os neurônios são treinados para responder uma cor que representa a posição do centro do campo receptivo do neurônio em relação ao centro da caixa delimitadora do objeto. Cada neurônio acrescenta sua própria descrição (par entrada-saída) do estado corrente da camada de entrada na memória compartilhada pelos neurônios.

O retreino, por outro lado, é automaticamente requerido pelo módulo Validation no momento em que uma nova aparência do objeto é observada. Para este novo treino, realizado a fim de reforçar a descrição atual do objeto no sistema, utilizamos as suposições anteriores do sistema sobre a localização do objeto de interesse, ou seja, os resultados dos movimentos sacádicos realizados em *frames* anteriores.

É necessário garantir uma boa escolha da aparência do objeto a ser utilizada no retreino do sistema. Para tal, inicialmente, consideramos uma janela temporal (últimos j *frames*, onde j é parâmetro do sistema) para escolher a melhor representação atual do objeto. Em seguida, realizamos um ajuste em torno do alvo sacádico do *frame* escolhido na janela temporal com o objetivo de evitar problemas de centralização. Neste ajuste, o centro de atenção é movido para cada um dos *pixels* de uma janela de $numPixels \times numPixels$ *pixels* (onde $numPixels$ é parâmetro do sistema) e a matriz acumuladora é recalculada para cada *pixel*. O *pixel* com o valor mais alto de confiança é escolhido como o centro do objeto de interesse que será utilizado para retreinar o sistema.

Com o propósito de evitar treinar o sistema com imagens impróprias do objeto de interesse, algumas restrições são impostas. A primeira restrição é uma confiança mínima para o retreino, ou seja, a confiança do *pixel* escolhido como centro do objeto de interesse deve ser maior do que um limiar parametrizado do sistema. A segunda restrição é o número mínimo de *frames* entre treinos, em que sistema só pode retreinar uma imagem após 10 *frames* desde o último treino. Esta restrição previne a saturação da memória da rede, evitando que muitos exemplos da mesma aparência do objeto sejam armazenados. Por fim, a última restrição é a posição espacial do objeto. O objeto de interesse não pode estar muito perto das bordas da

imagem durante o retreino, evitando o sistema aprender partes do objeto de interesse exterior a imagem de entrada.

Uma importante característica a ser analisada no retreino é o tamanho da memória compartilhada. No modelo proposto, foi utilizada uma camada neural de 65 x 48 neurônios e uma tabela-verdade (memória) com tamanho de $65 \times 48 \times 32$. Esta quantidade de entradas é equivalente a 32 linhas por neurônio na tabela-verdade, ou seja, o sistema tem memória para armazenar 32 representações completas do objeto de interesse. Quando a tabela-verdade está completa e o retreino é realizado, cada neurônio treinado substitui uma entrada da tabela-verdade aleatoriamente, acarretando uma degradação tênue dos conteúdos anteriores.

4.2 Tracking

O módulo Tracking é responsável por rastrear (seguir) o objeto de interesse *frame* a *frame*. Uma restrição imposta pelo módulo é que o objeto de interesse tem de ser visível no *frame* anterior, pois a localização do objeto no *frame* anterior é utilizada como centro de atenção inicial do *frame* atual. Para ser capaz de localizar o objeto no *frame* atual o objeto não pode mover-se para fora do campo visual definido pela camada de entrada.

4.2.1 Etapas de Tracking

O procedimento de rastreamento compreende 3 etapas: (i) a primeira sequência de movimentos sacádicos, (ii) o ajuste da escala e (iii) a segunda sequência de movimentos sacádicos.

Para permitir um melhor alinhamento do centro do objeto de interesse com o centro da camada de entrada, a sequência inicial de movimentos sacádicos pode compreender mais do que um único movimento sacádico. Uma vez que o centro do objeto de interesse se aproxima da região da fóvea do nosso sistema (do centro de

atenção), o mapeamento log-polar é mais preciso e, conseqüentemente, os movimentos sacádicos são mais curtos e precisos. A seqüência inicial de movimentos sacádicos é realizada até que não ocorra mudança na posição do centro de atenção ou até que um máximo de quatro movimentos sejam realizados.

Para garantir que a rede neural seja capaz de identificar um objeto de interesse corretamente, a imagem deste objeto deve ter tamanho semelhante na camada de entrada em todo o vídeo, mantendo o padrão de cor aprendido pelos neurônios. Portanto, o processo de rastreamento deve ajustar adequadamente ao longo do tempo o tamanho da imagem do objeto de interesse. Os pormenores do processo de ajuste de escala serão apresentados na subseção seguinte.

Para refinar a localização do centro do objeto de interesse, após o ajuste de escala, uma segunda seqüência de movimentos sacádicos é realizada.

Após todas as etapas do procedimento de rastreamento realizadas, o módulo responde (i) o centro de atenção que melhor representa o centro do objeto de interesse, (ii) a escala da imagem do objeto de interesse e (iii) a confiança do último movimento sacádico. A caixa delimitadora do objeto de interesse tem uma proporção fixa, sendo seu aspecto baseado na caixa delimitadora inicial do objeto de interesse manualmente anotada e seu tamanho baseado na fração da escala atual pela escala inicial do objeto.

4.2.2 Estimativa da Escala

O método mais comum na literatura para detectar um objeto de interesse em uma imagem em que a escala do objeto é desconhecida é tentar detectar tal objeto com todas as possíveis escalas nesta imagem. Embora esta abordagem traga bons resultados, ela exige a execução do processo de detecção muitas vezes, sendo muito dispendiosa. Para evitar tantas tentativas, propomos uma solução para estimar a escala de um objeto *frame a frame*.

O método proposto para o ajuste de escala segue a ideia da matriz acumuladora utilizada nos movimentos sacádicos, entretanto, ao invés de calcular um deslocamento espacial, um fator de escala é computado. Um vetor acumulador unidimensional é usado para computar votos para todas as possíveis escalas do objeto, onde a posição central do vetor representa o objeto na escala inicial (fator de escala igual a um) e as posições acima ou abaixo do seu centro aumentam ou diminuem o fator de escala em 0,02, respectivamente.

Cada neurônio ativado $n_{i,j}$ da camada neural contribui com um voto para o provável fator de escala $z_{i,j}$ do objeto, sendo este voto ponderado por uma medida de confiança da resposta do neurônio (o peso $w_{i,j}$ que foi apresentado no capítulo anterior). Os votos são armazenados no vetor acumulador de fatores de escala e um filtro gaussiano suaviza tais contribuições. Para escolher a escala atual do objeto de interesse, um procedimento “*winner-takes-it-all*” é executado sobre o vetor acumulador e o fator de escala selecionado é convertido na informação (i) a escala não mudou, (ii) a escala é 5% maior ou (iii) a escala é 5% menor do que o fator de escala do *frame* anterior. Se o fator de escala selecionado esta na posição central do vetor acumulador (ou em um intervalo parametrizado que compreende esta posição) é definido que a escala não mudou. Para fatores de escala acima ou abaixo deste intervalo, o acréscimo ou decréscimo de 5% do fator de escala atual é aplicado. A Figura 23 ilustra o mecanismo de ajuste de fator de escala automático.

O método proposto exige que o centro de atenção esteja posicionado no centro do objeto ou bem próximo a ele. Sendo assim, podemos supor que, se a cor de um neurônio ativo está na posição espacial correta na camada neural - isto é, na mesma posição correspondente a sua cor durante o treino inicial -, o vetor $L_{i,j}(r, \theta)$ deve ser igual ao vetor $C_{i,j}(r_c, g_c)$ (discutido no capítulo anterior), correspondendo a um fator de escala igual a um. Caso contrário, um vetor deve ser maior do que o outro e o fator de escala deve ser menor ou maior do que o anterior. Portanto, o fator de escala $z_{i,j}$ sugerido pela cor e posição do neurônio $n_{i,j}$ pode ser calculado pela fração da magnitude de um vetor pela magnitude do outro, como na equação abaixo:

$$z_{i,j} = |C_{i,j}(r_c, g_c)| / |L_{i,j}(r, \theta)|.$$

Uma vez que a parte central do objeto é representada pelo centro do mapeamento log-polar, mudanças de escala não podem ser adequadamente capturadas nesta região devido ao excesso de amplificação (ou amplificação infinita quando $|L_{i,j}(r, \theta)| = 0$). Portanto, a fim de evitar ruídos no fator de escala, a contribuição dos neurônios localizados no centro da camada neural (neurônios com campo receptivo interior a um raio de 5 pixels a partir do centro de atenção) não são adicionadas ao vetor acumulador (ver região verde na Figura 23).

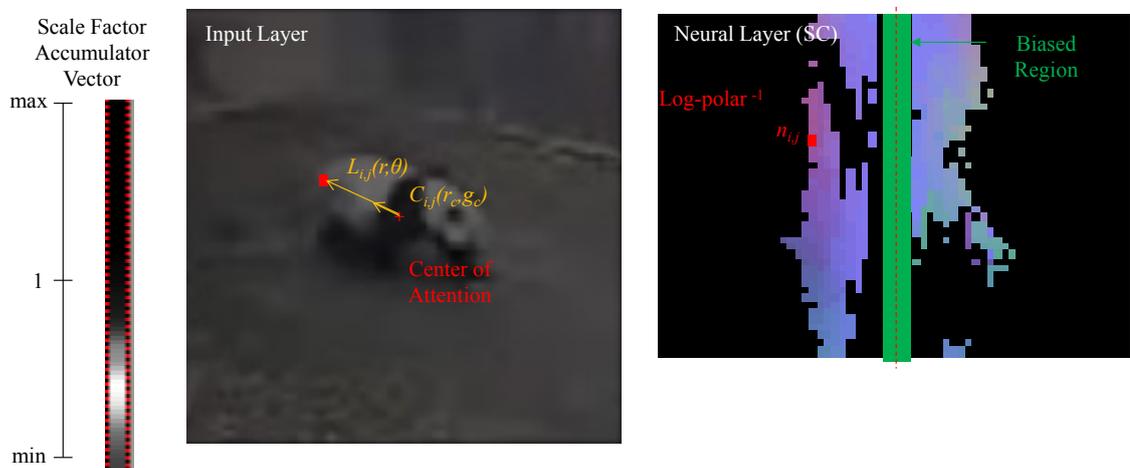


Figura 23: Ilustração do mecanismo de ajuste automático de escala. Cada posição no vetor acumulador de fatores de escala corresponde a uma escala indicada por $z_{i,j} = |C_{i,j}(r_c, g_c)| / |L_{i,j}(r, \theta)|$. Esta posição é incrementada de acordo com o peso correspondente $w_{i,j}$ de $n_{i,j}$. Um filtro gaussiano é utilizado para suavizar os dados no vetor acumulador. Figura retirada de [61].

A Figura 24 mostra exemplos de vetores acumuladores para objetos em diferentes escalas (maior, igual e menor) e um vídeo ilustrando o processo de zoom pode ser visto em <https://www.youtube.com/watch?v=UMDi88V4Vs>.

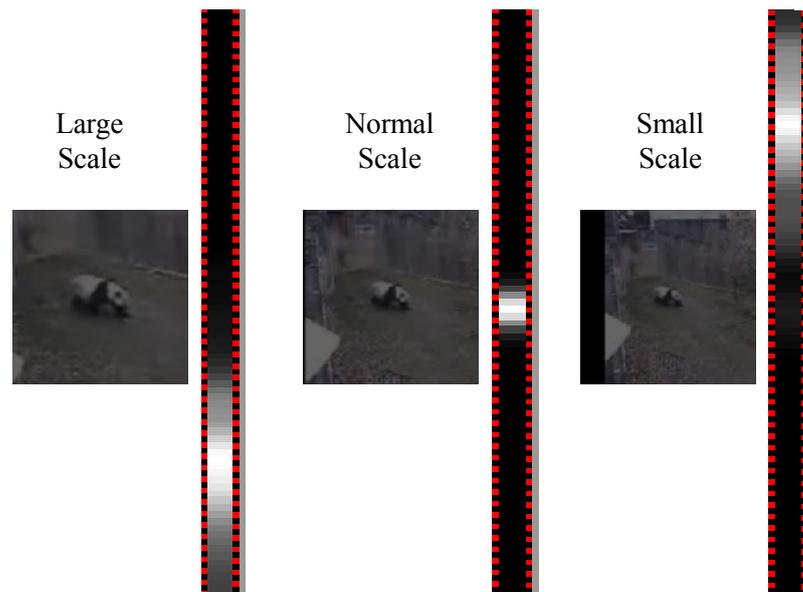


Figura 24: Objetos em diferentes escalas com seus respectivos estados do vetor acumulador de fatores de escala. Figura retirada de [61].

4.3 Detection

O módulo Detection é responsável por detectar o objeto de interesse no momento em que ele reaparece na imagem, ou seja, no momento em que, depois de ter sofrido oclusão, o objeto volta a ser visível. Uma vez que o campo de visão da camada de entrada pode não cobrir todo o *frame* (o seu tamanho varia de acordo com o fator de escala), pode não ser possível localizar o objeto com uma única sequência de movimentos sacádicos, sendo necessário explorar toda a imagem em busca do objeto.

A exploração da imagem do *frame* é realizada dividindo-o em sub-regiões a serem cobertas pelo campo visual definido pela camada de entrada. Durante a exploração, para cada sub-região, o centro de atenção inicial é o centro de cada sub-região, uma sequência de movimentos sacádicos é realizada e as confianças destes movimentos são armazenadas.

Uma vez que o objeto pode reaparecer em uma escala diferente do que a do momento em que saiu de cena, o mesmo procedimento de exploração da imagem é

repetido para cada possível escala do objeto (nos experimentos testamos o tamanho da caixa delimitadora compreendido no intervalo de 20 *pixels* até 5 vezes o tamanho inicial, incrementado em 10% em cada procedimento).

Depois de analisar todas as regiões da imagem e todas as possíveis escalas do objeto, o alvo do movimento sacádico com a maior confiança e sua escala associada é selecionado. Se a confiança estiver acima de um limiar parametrizado, o objeto foi detectado.

4.4 Validation

O módulo Validation é responsável por decidir qual será o próximo módulo do sistema a ser executado. Para tal decisão, é examinada a confiança do último movimento sacádico, a informação de qual foi o último módulo a operar e o estado atual do sistema.

O sistema possui dois estados: “o objeto é visível” e “o objeto não é visível”. Se a confiança do último movimento sacádico for menor que um limiar parametrizado Ψ , o módulo Validation deverá mudar o estado do sistema para “o objeto não é visível” e nenhuma caixa delimitadora deverá ser exibida. Caso contrário, o módulo deverá mudar o estado para “o objeto é visível” e uma caixa delimitadora centralizada no centro de atenção e dimensionada de acordo com a escala atual deverá ser traçada.

Para cada novo *frame* do vídeo, o módulo Validation pode decidir de diversas formas, apresentadas a seguir. Após a operação do módulo (Re)Learning, o módulo Validation ativa o módulo Tracking. Após a operação do módulo Tracking, o módulo Validation pode: (i) ativar o módulo Tracking novamente, caso o objeto seja visível; (ii) ativar o módulo Detection, caso o objeto não seja visível; ou (iii) ativar o módulo (Re)Learning, caso o objeto seja visível e a confiança do movimento sacádico estiver abaixo do limiar parametrizado de retreino, Ω . Uma importante observação é que o limiar de retreino (Ω) é bem maior do que o limiar que indica que o objeto não está presente na cena (Ψ). Após a operação do módulo Detection, o módulo Validation

pode ativar o mesmo módulo de novo, caso o objeto não seja visível, ou ativar o módulo Tracking, caso o objeto seja visível. Esta última transição requer uma confiança maior do que o limiar de retreino Ω , assegurando que o objeto encontrado é de fato o objeto de interesse.

5 AVALIAÇÃO EXPERIMENTAL E RESULTADOS

Neste capítulo apresentamos a avaliação experimental do sistema proposto e os resultados obtidos. Foram realizados três experimentos: o experimento *dataset* TLD, o experimento siga-o-líder e o experimento *eye-tracker*. O objetivo do primeiro experimento é comparar o sistema proposto com outro método conhecido de rastreamento de objetos de interesse. O objetivo do segundo experimento é avaliar a capacidade do IARA, o carro autônomo do LCAD-UFES, de seguir um objeto de interesse (carro) utilizando o sistema. Por fim, o objetivo do terceiro experimento é comparar o sistema proposto com o sistema visual humano.

5.1 Experimento *Dataset* TLD

Nesta seção apresentamos a metodologia, a métrica, a calibração da rede e os resultados alcançados do experimento *dataset* TLD.

5.1.1 Metodologia

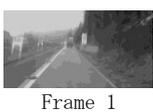
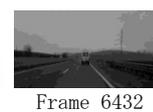
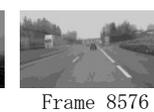
Nesta subseção, apresentamos a metodologia utilizada na implementação da arquitetura proposta neste trabalho.

5.1.1.1 *Dataset* TLD

Para avaliar o desempenho do sistema de rastreamento visual, utilizamos o *dataset* TLD (disponível em <http://personal.ee.surrey.ac.uk/Personal/Z.Kalal/tld.html>), um importante *dataset* de avaliação de sistemas de rastreamento de objetos. O *dataset* TLD é composto por 10 *benchmarks* com vídeos que contém tipos de objetos diferentes.

O vídeo David possui 761 *frames* e mostra uma pessoa (objeto de interesse) andando por uma sala que encontra-se inicialmente escura e, após alguns *frames*, torna-se iluminada. O vídeo Jumping tem 313 *frames* e apresenta uma pessoa pulando corda. Os vídeos Pedestrian1 (140 *frames*), Pedestrian2 (338 *frames*) e Pedestrian3 (184 *frames*) mostram pedestres andando em diferentes ambientes. Os vídeos Car (945 *frames*) e Volkswagen (8.576 *frames*) apresentam carros em movimento. O vídeo Motocross possui 2.665 *frames* de uma corrida de moto. O vídeo Carchase tem 9.928 *frames* e mostra um carro que está sendo perseguido por um carro da polícia filmados de um helicóptero. O vídeo Panda possui 3.000 *frames* e apresenta um urso panda andando por um ambiente de um zoológico. A Tabela 2 mostra alguns *frames* dos vídeos do *dataset* TLD.

Tabela 2: Exemplos de frames dos vídeos do dataset TLD.

David (1)					
	Frame 1	Frame 190	Frame 380	Frame 570	Frame 761
Jumping (2)					
	Frame 1	Frame 78	Frame 156	Frame 234	Frame 313
Pedestrian 1 (3)					
	Frame 1	Frame 35	Frame 70	Frame 105	Frame 140
Pedestrian 2 (4)					
	Frame 1	Frame 84	Frame 169	Frame 253	Frame 338
Pedestrian 3 (5)					
	Frame 1	Frame 46	Frame 92	Frame 138	Frame 184
Car (6)					
	Frame 1	Frame 236	Frame 472	Frame 708	Frame 945
Motocross (7)					
	Frame 1	Frame 666	Frame 1332	Frame 1998	Frame 2665
Volkswagen (8)					
	Frame 1	Frame 2144	Frame 4288	Frame 6432	Frame 8576
Carchase (9)					
	Frame 1	Frame 2482	Frame 4964	Frame 7446	Frame 9928
Panda (10)					
	Frame 1	Frame 750	Frame 1500	Frame 2250	Frame 3000

Dentre as diversas condições desafiadoras do *dataset* TLD, podemos citar movimentos abruptos da câmera, mudanças de escala/pose/iluminação e oclusões

parciais e totais. A Tabela 3 sumariza as propriedades de cada vídeo. Os vídeos são manualmente anotados e possuem um único objeto de interesse.

Tabela 3: Propriedades do dataset TLD.

<i>Propriedades</i>	<i>Vídeos</i>									
	1	2	3	4	5	6	7	8	9	10
Câmera em movimento	sim	sim	sim	sim	sim	sim	sim	Sim	sim	sim
Oclusão Parcial	sim	não	não	sim						
Oclusão Total	não	não	não	sim						
Mudança de Posição	sim	não	não	não	não	não	sim	Sim	sim	sim
Mudança de Iluminação	sim	não	não	não	não	não	sim	Sim	sim	sim
Mudança de Escala	sim	não	não	não	não	não	sim	Sim	sim	sim
Objetos Similares	não	não	não	sim	sim	sim	sim	Sim	sim	não

5.1.1.2 Framework de Inteligência Artificial MAE

Para implementar o sistema de rastreamento visual, utilizamos o *framework* de inteligência artificial MAE (Máquina Associadora de Eventos) [63]. O *framework* MAE possui código fonte aberto e foi desenvolvido pelo Grupo de Pesquisa em Ciência da Cognição do LCAD–UFES [64]. Com este *framework*, pode-se projetar tanto estruturas modulares usando RNSP com neurônios do tipo VG-RAM [54] quanto estruturas com arquitetura em camadas, com a definição de um processamento específico para cada camada [65]. Várias são as facilidades fornecidas pelo *framework*, tais como facilidade de programação, flexibilidade, facilidade visual e portabilidade.

O sistema MAE é composto, basicamente, por duas partes. A primeira parte é um programa chamado netcomp, que recebe como entrada um arquivo com extensão .con contendo a descrição da arquitetura (neural ou de camadas) e gera como saída um arquivo com extensão .c, contendo código C correspondente à tradução do arquivo .con para C. A segunda parte é composta por bibliotecas de rotinas MAE, que implementam os neurônios, seu treinamento, processamento específico para cada camada, interface com o usuário, etc.

O netcomp é um tradutor da linguagem de descrição da arquitetura definida no arquivo .con para C. O arquivo .c gerado e os arquivos de rotinas do usuário são compilados e linkados com bibliotecas de rotinas MAE necessárias para um *script* chamado netcompiler. É este *script* que gera o arquivo executável final, ou aplicação

MAE, que implementa a arquitetura descrita no arquivo .con, além de uma interface que permite ao usuário manipular a arquitetura. Na verdade, para gerar uma aplicação MAE, este processo é transparente ao usuário (Figura 25). O netcompiler se encarrega de realizar todo o processo automaticamente, gerando como saída uma aplicação MAE. Maiores detalhes podem ser encontrados em [65].

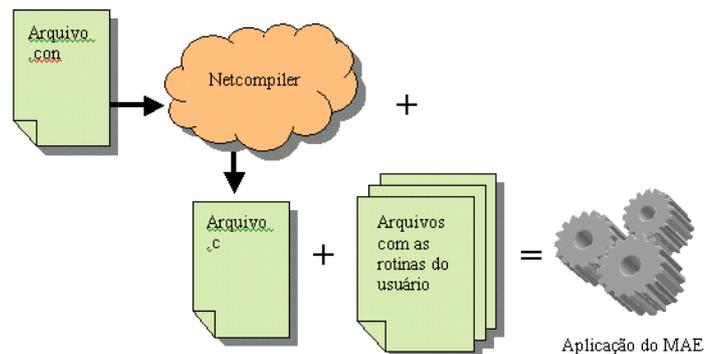


Figura 25: Esquema de criação de uma aplicação utilizando o framework MAE. Figura retirada de [53].

5.1.2 Métrica

Nesta subseção apresentamos a métrica utilizada para avaliar o sistema de rastreamento visual proposto. Realizamos experimentos com todos os vídeos do banco de dados TLD. A caixa delimitadora manualmente anotada do primeiro *frame* foi utilizada para realizar o treino inicial do sistema. Após o processamento de cada vídeo, todas as ocorrências de verdadeiros positivos (TP), falsos positivos (FP) e *frames* com objetos de interesse (FWO) foram contabilizados. Verdadeiros positivos acontecem quando, nos *frames* em que as informações do banco de dados indicam a presença do objeto de interesse, o sistema **aponta corretamente para o objeto**. Falsos positivos acontecem quando: (i) nos *frames* em que as informações do banco de dados indicam a ausência do objeto de interesse, o sistema incorretamente responde com uma caixa delimitadora; ou, (ii) nos *frames* em que as informações do banco de dados indicam a presença do objeto de interesse, o sistema não **aponta corretamente para o objeto**. *Frames* com objetos de interesse são *frames* em que informações do banco de dados indicam a presença do objeto de interesse.

A decisão sobre se o sistema **aponta corretamente para o objeto** ou não foi tomada a partir do coeficiente de Jaccard [66]. O coeficiente de Jaccard mede a similaridade entre conjuntos de amostras finitas e é definido como sendo a interseção dividida pela união dos conjuntos de amostras (Equação 5).

$$\text{jaccard} = \frac{B1 \cap B2}{B1 \cup B2} = \frac{I}{(B1+B2)-I} \quad \text{Equação 5}$$

O coeficiente de Jaccard pode ser usado para medir a sobreposição entre as duas caixas delimitadoras (a encontrada pelo sistema e a fornecida pela base de dados). Na Figura 26, B1 é a área da primeira caixa delimitadora, B2 a área da segunda caixa delimitadora e I é a área da interseção entre as duas caixas delimitadoras. O coeficiente de Jaccard entre as duas caixas delimitadoras é obtido dividindo-se a área da interseção pela área da união (Equação 5). Como em [67], o sistema **aponta corretamente para o objeto** se o coeficiente de Jaccard for maior do que 0,25.

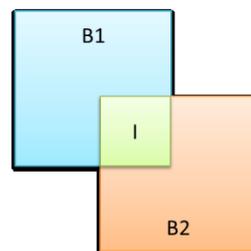


Figura 26: Ilustração do coeficiente de Jaccard.

Com base nas medidas acima descritas, avaliamos o desempenho do sistema por meio de três métricas: Precision (P), Recall (R) e F-measure (F). Precision é o número de verdadeiros positivos dividido pelo número de todas as respostas positivas dos sistema ($P = TP / (TP + FP)$). Recall é o número de verdadeiros positivos dividido pelo número de ocorrência de objetos de interesse ($R = TP / FWO$). F-measure é a média harmônica de Precision e Recall ($F = 2 * P * R / (P + R)$).

5.1.3 Calibração do Sistema

Nesta subseção apresentamos a calibração dos parâmetros do sistema de busca visual e do sistema de rastreamento visual propostos.

5.1.3.1 Calibração da Busca Visual

Buscando obter o melhor resultado nos experimentos executados neste capítulo, realizamos a calibração dos parâmetros da RNSP VG-RAM utilizada no sistema de busca visual. O objetivo principal foi alcançar a configuração paramétrica que obtivesse a melhor precisão na tarefa de encontrar um objeto de interesse, ou seja, na busca visual. Utilizamos, para a calibração, 100 imagens do banco de dados do TLD, sendo 10 imagens aleatórias de cada *benchmark*.

Como apresetado anteriormente, a RNSP WNN possui 4 parâmetros que devem ser calibrados: a dimensão da camada neural ($m \times n$), o número de sinapses por neurônio (w), o desvio padrão da distribuição normal (σ) e o fator de log utilizado na função log-polar (α). A Tabela 4 apresenta o conjunto de valores testados para cada parâmetro, que resultam em 720 configurações distintas da rede.

Tabela 4: Conjunto de valores testados para cada parâmetro da rede.

<i>Parâmetros</i>	<i>Valores Testados</i>
Dimensão da camada neural ($m \times n$)	5x4, 9x8, 17x12, 65x48 e 129x96
Número de sinapses por neurônio	32, 64, 128 e 256
Desvio padrão da distribuição normal	2, 4, 6, 8, 10 e 12
Fator de log	2, 4, 6, 8, 10 e 12

Foram realizados dois experimentos de calibração: o primeiro baseado na confiança da rede após um movimento sacádico e o segundo baseado no resultado do movimento sacádico. Os experimentos são apresentados a seguir.

5.1.3.1.1 Calibração Baseada na Confiança da Rede

O processo de calibração baseado na confiança da rede é descrito a seguir. Inicialmente, o sistema obtém uma imagem do banco de dados e realiza, a partir dos dados do *ground truth* disponibilizados, o treinamento da RNSP. Após o treinamento, o sistema direciona o centro de atenção para uma região fora da caixa delimitadora do objeto de interesse, embora dentro da janela de atenção, e uma sacada é então realizada. Por fim, o centro de atenção é deslocado para fora da janela de atenção (o objeto de interesse não será visualizado) e uma segunda sacada é realizada. Uma importante observação é que é realizado um novo treinamento a cada imagem em cada *benchmark* e a memória anterior é esquecida (apagada).

As sacadas realizadas no processo de calibração geram *tuplas* contendo o número da imagem, a escala, as coordenadas do alvo sacádico, as confianças obtidas nas sacadas, a dimensão da camada neural, o número de sinapses por neurônio, o desvio padrão da distribuição normal, o fator de log utilizado na função log-polar, o índice de similaridade de jaccard e o fator obtido pela equação abaixo:

$$fator = \frac{C1 - C2}{C1}$$

onde C1 é confiança obtida na sacada próxima ao objeto de interesse (sacada 1) e C2 é a confiança obtida na sacada longe do objeto de interesse (sacada 2). Este fator foi utilizado como métrica de desempenho da rede.

5.1.3.1.1.1 Variação da Dimensão da Rede

Nesta seção apresentamos os resultados do desempenho da RNSP em função da quantidade de neurônios na rede.

O gráfico da Figura 27 apresenta as mudanças ocorridas no desempenho da RNSP ao longo do crescimento da dimensão da rede em relação às configurações

do conjunto de parâmetros. Os vértices presentes em cada curva representam o desempenho alcançado (eixo Y) para cada dimensão da camada neural (eixo X). É possível perceber que há um ganho significativo no desempenho da rede quando a dimensão da rede ($m \times n$) alcança 65×48 neurônios, independente da configuração dos demais parâmetros. Após este ápice, o desempenho da rede volta a cair.

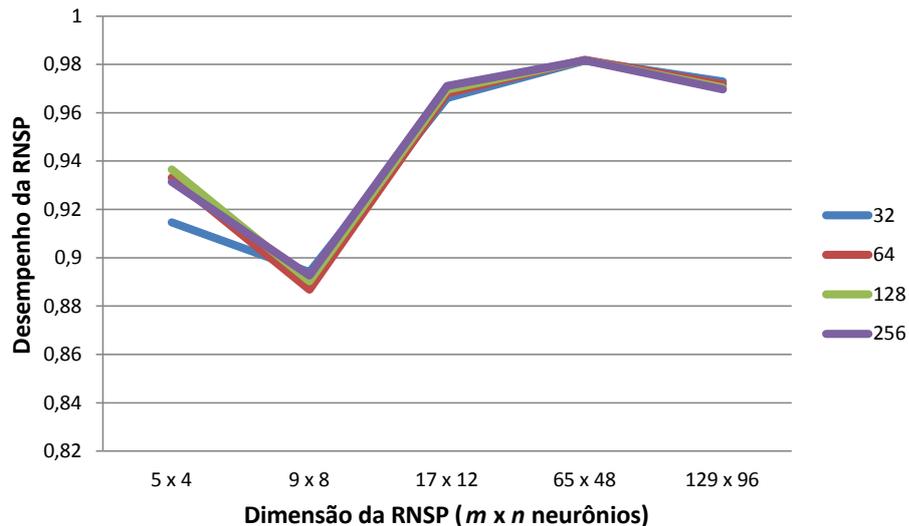


Figura 27: Gráfico que apresenta as mudanças ocorridas no desempenho das RNSP ao longo do crescimento da dimensão da rede.

A Tabela 5 representa numericamente os valores utilizados para representar o gráfico da Figura 27.

Tabela 5: Mudanças ocorridas no desempenho da RNSP ao longo do crescimento da dimensão da rede.

		Número de Sinapses			
		32	64	128	256
Dimensão da Camada Neural	5 x 4	0,91	0,93	0,93	0,93
	9 x 8	0,89	0,88	0,89	0,89
	17 x 12	0,96	0,96	0,96	0,97
	65 x 48	0,98	0,98	0,98	0,98
	129 x 96	0,97	0,97	0,97	0,96

5.1.3.1.1.2 Variação do Número de Sinapses

Nesta seção apresentamos os resultados do desempenho da RNSP em função do número de sinapses por neurônio. Para gerar estes resultados utilizamos, estaticamente, os melhores resultados observados para os demais parâmetros de

entrada da rede (2 para o desvio padrão da distribuição normal e 10 para o fator de escala da função log-polar).

O gráfico da Figura 28 ilustra a tendência da RNSP VG-RAM para o conjunto de variações das sinapses dos neurônios durante a calibração. Os vértices presentes na curva representam a média de desempenho obtido por todas as dimensões testadas da camada neural (eixo Y) para cada número de sinapses (eixo X). Observamos que a maior média foi alcançada com 256 sinapses.

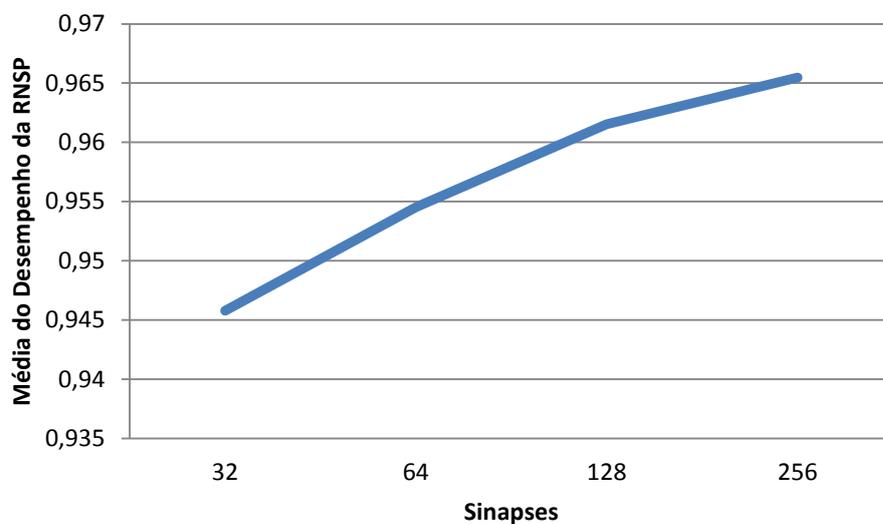


Figura 28: Gráfico de tendências de desempenho da variação de sinapses em função da média do desempenho das dimensões testadas da RNSP.

5.1.3.1.1.3 Variação do Desvio Padrão da Distribuição Normal

Nesta seção apresentamos os resultados da calibração do sistema variando o desvio padrão da distribuição normal gaussiana aplicada no padrão de interconexão das sinapses dos neurônios em função da dimensão da rede. Para gerar estes resultados utilizamos, estaticamente, os melhores resultados observados para os demais parâmetros de entrada da rede (2 para o fator de escala da função log-polar e 256 sinapses).

O gráfico da Figura 29 mostra a tendência da rede para o conjunto de variações do desvio padrão da distribuição normal das sinapses. Os vértices presentes na curva representam a média de desempenho obtido por todas as dimensões testadas

da camada neural (eixo Y) para cada desvio padrão (eixo X). É possível perceber que a rede atinge sua maior média de desempenho com o valor de desvio padrão da distribuição normal 10.

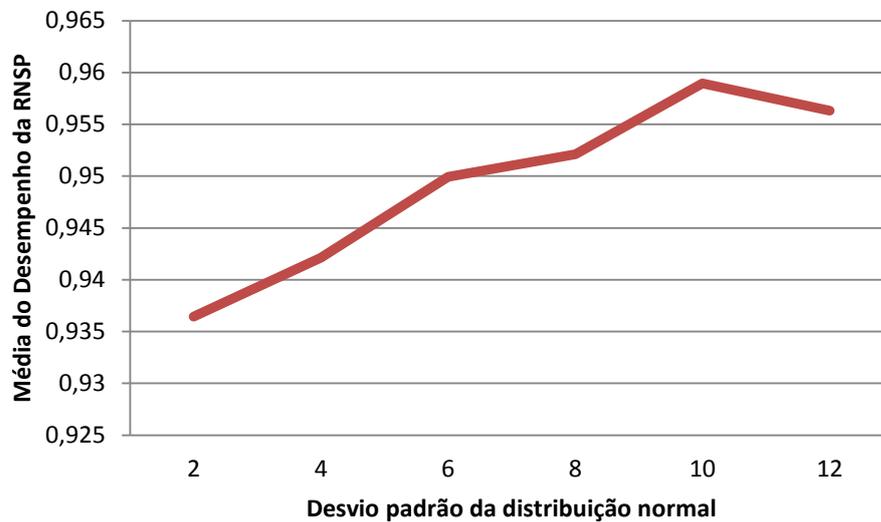


Figura 29: Gráfico de tendências de desempenho da variação do desvio padrão da distribuição normal em função da média do desempenho das dimensões testadas da RNSP.

5.1.3.1.1.4 Variação do Fator de Log

Nesta seção apresentamos os resultados da calibração do sistema variando o fator de log (fator de ampliação) da transformação de redimensionamento sofrida pela imagem na RNSP. Para gerar estes resultados utilizamos, estaticamente, os melhores resultados observados para os demais parâmetros de entrada da rede (10 de desvio padrão da distribuição normal (distribuição das sinapses) e 256 sinapses).

A Figura 30 ilustra as tendências de desempenho para o conjunto de variações do fator de log relativo ao fator de escala, utilizados na função log-polar, responsável pelo mapeamento retinotópico da imagem de entrada para a rede. Os vértices presentes na curva representam a média de desempenho obtido por todas as dimensões testadas da camada neural (eixo Y) para cada valor de log-polar (eixo X). É possível observar que os menores valores para o fator de escala contribuem para um resultado melhor.

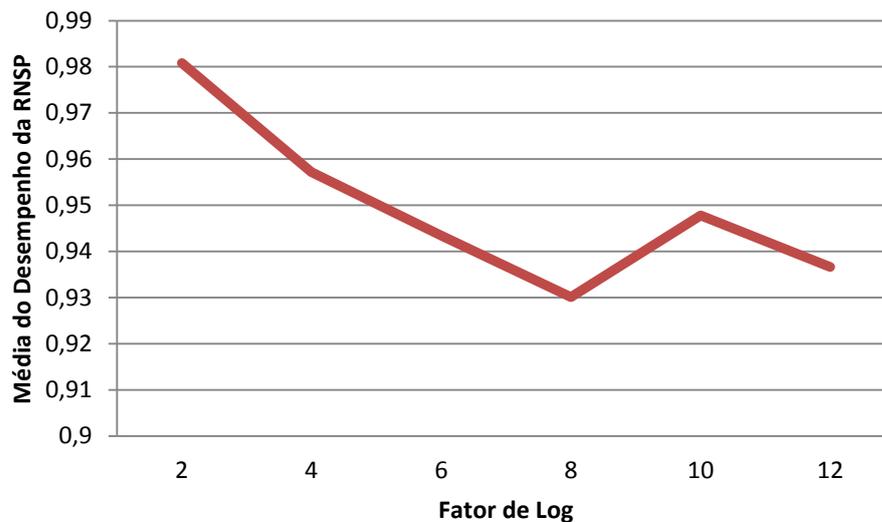


Figura 30: Gráfico de tendências de desempenho da variação do fator de log em função da média de desempenho das dimensões testadas da RNSP.

5.1.3.1.2 Calibração Baseada no Resultado do Movimento Sacádico

O processo de calibração baseado no resultado da sacada é descrito a seguir. Inicialmente, o sistema obtém uma imagem do banco de dados e realiza, a partir dos dados do *ground truth* disponibilizado, o treinamento da RNSP. Após o treinamento, o sistema direciona o centro de atenção para uma região fora da caixa delimitadora do objeto de interesse, embora dentro da janela de atenção, e uma sacada é então realizada. Para cada sacada é analisado se o sistema aponta corretamente para o objeto através do coeficiente de jaccard, que deve ser maior do que 0,75. Uma importante observação é que é realizado apenas um treinamento para cada *benchmark*.

As sacadas realizadas no processo de calibração geram *tuplas* contendo o número da imagem, a escala, as coordenadas do alvo sacádico, as confianças obtidas nas sacadas, a dimensão da camada neural, o número de sinapses por neurônio, o desvio padrão da distribuição normal, o fator de log utilizado na função log-polar e o índice de similaridade de jaccard. O índice de similaridade de jaccard é utilizado para sabermos se o sistema aponta corretamente para o objeto, e esta informação foi utilizado como métrica de desempenho da rede.

5.1.3.1.2.1 Variação da Dimensão da Rede

Nesta seção apresentamos os resultados do desempenho da RNSP em função da quantidade de neurônios na rede.

O gráfico da Figura 31 apresenta as mudanças ocorridas no desempenho da RNSP ao longo do crescimento da dimensão da rede em relação às configurações do conjunto de parâmetros. Os vértices presentes em cada curva representam o desempenho alcançado (eixo Y) para cada dimensão da camada neural (eixo X). É possível perceber que há um ganho significativo no desempenho da rede quando a dimensão da rede ($m \times n$) alcança 65×48 neurônios, independente da configuração dos demais parâmetros.

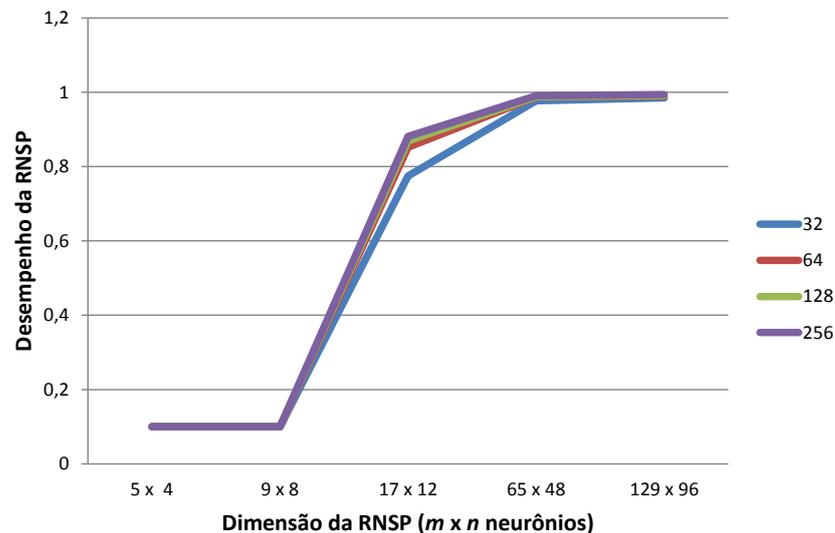


Figura 31: Gráfico que apresenta as mudanças ocorridas no desempenho da RNSP ao longo do crescimento da dimensão da RNSP.

A Tabela 6 representa numericamente os valores utilizados para representar o gráfico da Figura 31.

Tabela 6: Mudanças ocorridas no desempenho da RNSP ao longo do crescimento da dimensão da rede.

		Número de Sinapses			
		32	64	128	256
Dimensão da Camada Neural	5 x 4	0,1	0,1	0,1	0,1
	9 x 8	0,1	0,1	0,1	0,1
	17 x 12	0,77	0,85	0,86	0,88
	65 x 48	0,97	0,98	0,98	0,99
	129 x 96	0,98	0,98	0,99	0,99

5.1.3.1.2.2 Variação do Número de Sinapses

Nesta seção apresentamos os resultados da variação do número de sinapses por neurônio em função da quantidade de neurônios na rede. Para gerar estes resultados utilizamos, estaticamente, os melhores resultados observados para os demais parâmetros de entrada da rede (10 para o desvio padrão da distribuição normal e 2 para o fator de escala da função log-polar).

O gráfico da Figura 32 ilustra as tendências da RNSP VG-RAM para o conjunto de variações das sinapses dos neurônios durante a calibração. Os vértices presentes na curva representam a média de desempenho obtido por todas as dimensões da camada neural (eixo Y) para cada variação de sinapses (eixo X). Observamos que a maior média foi alcançada com 256 sinapses.

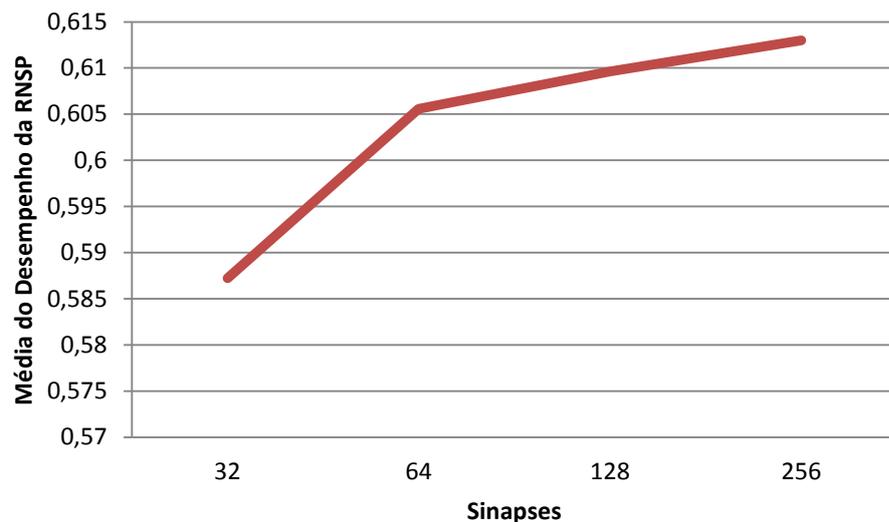


Figura 32: Gráfico de tendências de desempenho da variação das sinapses em função da média do desempenho das dimensões testadas da RNSP.

5.1.3.1.2.3 Variação do Desvio Padrão da Distribuição Normal

Nesta seção apresentamos os resultados da calibração do sistema variando o desvio padrão da distribuição normal gaussiana aplicada no padrão de interconexão das sinapses dos neurônios em função da dimensão da rede. Para gerar estes resultados utilizamos, estaticamente, os melhores resultados observados para os demais parâmetros de entrada da rede (2 para o fator de escala da função log-polar e 256 sinapses).

O gráfico da Figura 33 mostra a tendência da rede para o conjunto de variações do desvio padrão da distribuição normal das sinapses. Os vértices presentes na curva representam a média de desempenho obtido por todas as dimensões da camada neural (eixo Y) para cada desvio padrão (eixo X). É possível perceber que a rede atinge sua maior média de desempenho com o valor de desvio padrão da distribuição normal 10.

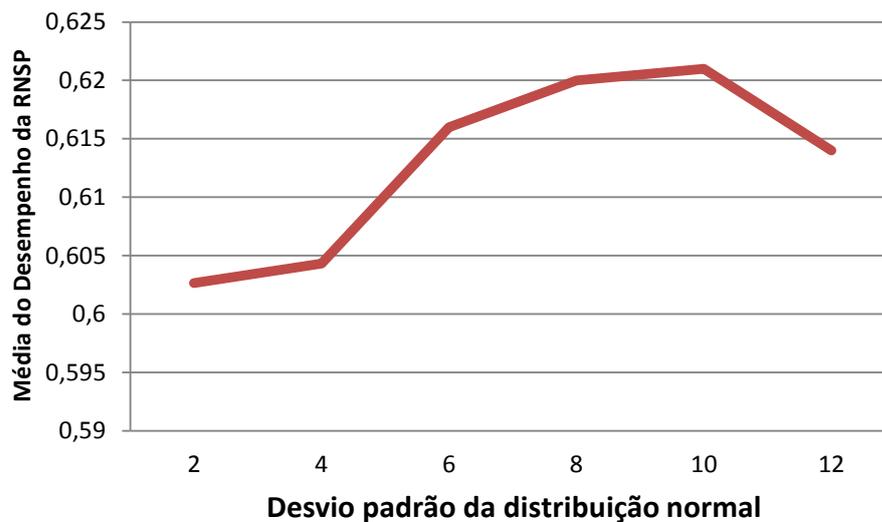


Figura 33: Gráfico de tendências de desempenho da variação do desvio padrão da distribuição normal em função da média do desempenho das dimensões testadas da RNSP.

5.1.3.1.2.4 Variação do Fator de Log

Nesta seção apresentamos os resultados da calibração do sistema variando o fator de log (fator de ampliação) da transformação de redimensionamento sofrida pela imagem na RNSP. Para gerar estes resultados utilizamos, estaticamente, os melhores resultados observados para os demais parâmetros de entrada da rede (10 de desvio padrão da distribuição normal (distribuição das sinapses) e 256 sinapses).

A Figura 34 ilustra as tendências de desempenho para o conjunto de variações do fator de log relativo ao fator de escala, utilizados na função log-polar, responsável pelo mapeamento retinotópico da imagem de entrada para a rede. Os vértices presentes na curva representam a média de desempenho obtido por todas as dimensões da camada neural (eixo Y) para cada valor de log-polar (eixo X). É

possível observar que o menor valor para o fator de escala contribui para um resultado melhor.

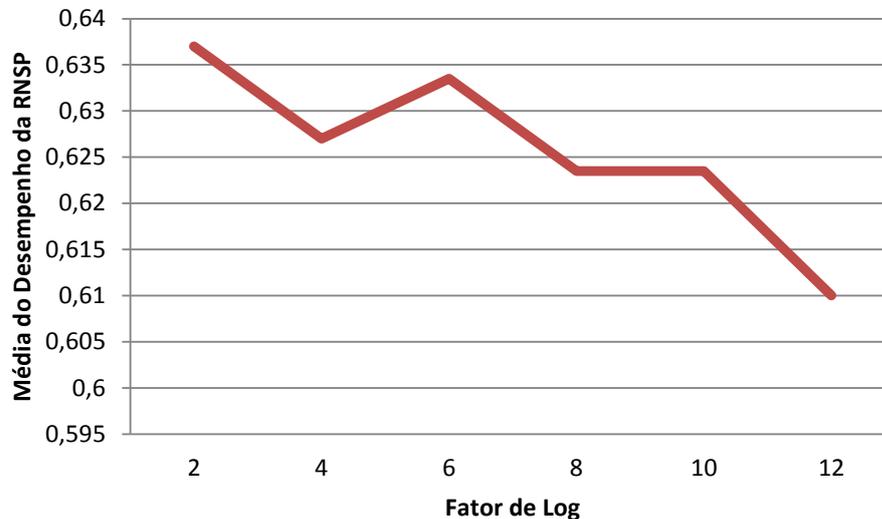


Figura 34: Gráfico de tendências de desempenho da variação do fator de log em função da média do desempenho das dimensões testadas da RNSP.

5.1.3.2 Calibração do Sistema de Rastreamento de Objetos

Buscando obter o melhor resultado nos experimentos executados neste capítulo, realizamos também a calibração dos parâmetros do sistema de rastreamento de objetos proposto. O objetivo principal foi alcançar a configuração paramétrica que obtivesse a melhor precisão na tarefa de rastrear um objeto a longo prazo. Utilizamos, para tal, o banco de dados do TLD e os melhores resultados observados para os parâmetros de entrada da rede (65 × 48 neurônios, 256 sinapses, 10 para o desvio padrão da distribuição normal e 2 para o fator de escala da função log-polar).

Como apresentado anteriormente, o sistema de rastreamento de objetos proposto possui 5 parâmetros que devem ser calibrados: o limiar de retreino (Ω), o parâmetro s utilizado no cálculo do peso dos neurônios, o número de *frames* utilizados na janela temporal para escolher a melhor representação atual do objeto no processo de retreino (j), o número de *pixels* utilizados no ajuste em torno do alvo sacádico do *frame* escolhido na janela temporal com o objetivo de evitar problemas

de centralização no processo de retreino (*numPixels*) e o tamanho da memória dos neurônios. A Tabela 7 apresenta o conjunto de valores testados para cada parâmetro.

Tabela 7: Conjunto de valores testados para cada parâmetro do sistema.

<i>Parâmetros</i>	<i>Valores Testados</i>
Limiar de retreino	20, 30, 35, 40, 45, 50, 55, 60, 65, 70, 80, 90, 100
Peso dos neurônios	1, 2, 3, 4, 5
Número de <i>frames</i> utilizados na janela temporal	1, 2, 3, 4, 5
Número de <i>pixels</i> utilizados no ajuste do alvo sacádico	1, 2, 3, 4, 5
Memória dos neurônios	1, 4, 8, 16, 32, 64 e 128

Executamos os experimentos deste trabalho em uma máquina Intel Core i7-4770 *quad-core* de 3,4 GHz e 16 GB de memória RAM. Utilizamos o sistema operacional Ubuntu 12.04 e o compilador gcc (GNU C Compiler) versão 4.4.1-2. Realizamos estes experimentos com 8 threads de OpenMP, por ser o número máximo de threads suportadas simultaneamente em hardware pelo processador adotado nos experimentos. O código C foi compilado com a seguinte sequência de flags: `-O3 -fopenmp -mpopcnt -msse2 -msse4.2 -ffast-math -mtune=native -march=native`.

Testamos os parâmetros utilizando os dois tipos de neurônios: VG-RAM e VG-RAM *fat-fast*. Os resultados obtidos são apresentados a seguir.

5.1.3.2.1 *Variação do Limiar de Retreino*

Nesta seção apresentamos os resultados do desempenho do sistema de rastreamento visual de objetos de interesse em função do limiar utilizado para retreino do sistema. Para gerar estes resultados utilizamos, estaticamente, o tamanho da memória dos neurônios igual a 32 e os demais parâmetros do sistema de rastreamento (o parâmetro s utilizado no cálculo do peso dos neurônios, o número de *frames* utilizados na janela temporal para escolher a melhor representação atual do objeto no processo de retreino (j), o número de *pixels* utilizados no ajuste em torno do alvo sacádico do *frame* escolhido na janela temporal com o objetivo de evitar problemas de centralização no processo de retreino (*numPixels*)) iguais a 3.

O gráfico da Figura 35 apresenta os resultados obtidos com neurônios do tipo VG-RAM com os limiares de retreino 20, 30, 35, 40, 45, 50, 55, 60, 65, 70, 80, 90 e 100. Os vértices presentes em cada curva representam o desempenho alcançado (eixo Y) para cada limiar de retreino (eixo X). É possível perceber que há um ganho significativo no desempenho da rede quando o limiar de retreino é igual a 55. Após este ápice, o desempenho da rede volta a cair.

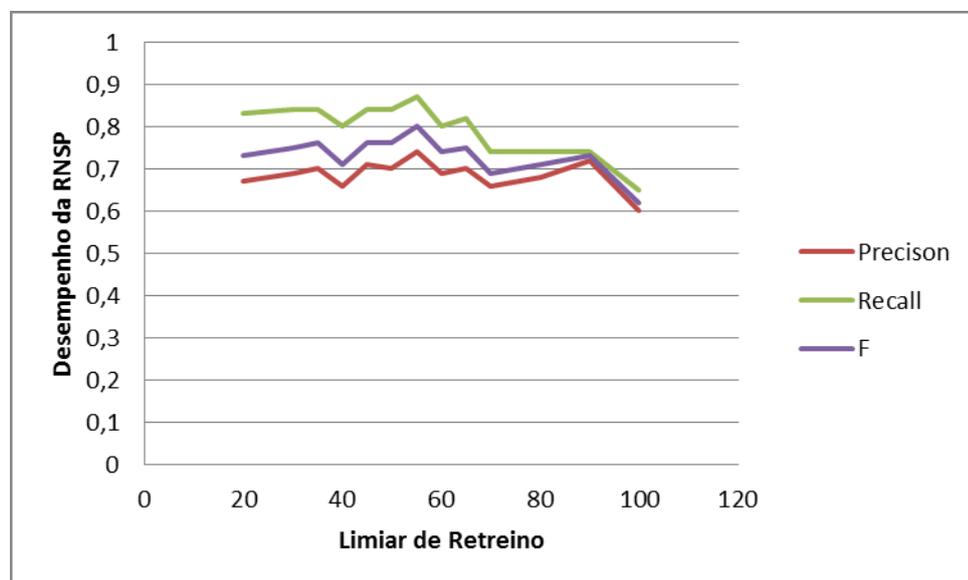


Figura 35: Gráfico que apresenta os resultados do desempenho da RNSP com neurônios do tipo VG-RAM em função dos limiares utilizado para retreino do sistema.

As tabelas 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19 e 20 apresentam numericamente os valores utilizados para representar o gráfico da Figura 35.

Tabela 8: Resultados obtidos com neurônios do tipo VG-RAM com limiar de retreino igual a 20.

Base	Precision	Recall	F
1	0,97	0,97	0,97
2	1,00	1,00	1,00
3	0,96	0,96	0,96
4	0,91	0,93	0,92
5	0,83	0,95	0,89
6	0,90	0,98	0,936
7	0,64	0,80	0,712
8	0,54	0,894	0,673
9	0,771	0,85	0,809
10	0,468	0,483	0,475
Media	0,67	0,83	0,73

Tabela 9: Resultados obtidos com neurônios do tipo VG-RAM com limiar de retreino igual a 30.

Base	Precision	Recall	F
1	0,89	0,89	0,89
2	1,00	1,00	1,00
3	0,84	0,84	0,84
4	0,95	0,95	0,95
5	0,87	0,97	0,92
6	0,89	0,97	0,931
7	0,68	0,82	0,741
8	0,529	0,866	0,657
9	0,763	0,836	0,798
10	0,68	0,68	0,68
Media	0,69	0,84	0,75

Tabela 10: Resultados obtidos com neurônios do tipo VG-RAM com limiar de retreino igual a 35.

Base	Precision	Recall	F
1	0,97	0,97	0,97
2	1,00	1,00	1,00
3	0,86	0,86	0,86
4	0,96	0,96	0,96
5	0,82	0,92	0,87
6	0,91	0,99	0,952
7	0,64	0,74	0,684
8	0,534	0,844	0,654
9	0,793	0,855	0,823
10	0,711	0,734	0,722
media	0,70	0,84	0,76

Tabela 11: Resultados obtidos com neurônios do tipo VG-RAM com limiar de retreino igual a 40.

Base	Precision	Recall	F
1	0,89	0,89	0,89
2	1,00	1,00	1,00

3	0,81	0,81	0,81
4	0,97	0,97	0,97
5	0,83	0,95	0,89
6	0,56	0,59	0,575
7	0,76	0,87	0,812
8	0,492	0,814	0,613
9	0,773	0,837	0,803
10	0,538	0,538	0,538
media	0,66	0,80	0,71

Tabela 12: Resultados obtidos com neurônios do tipo VG-RAM com limiar de retreino igual a 45.

Base	Precision	Recall	F
1	0,95	0,95	0,95
2	1,00	1,00	1,00
3	0,85	0,85	0,85
4	0,96	0,96	0,96
5	0,80	0,94	0,86
6	0,90	0,99	0,942
7	0,68	0,78	0,726
8	0,526	0,845	0,648
9	0,792	0,843	0,817
10	0,755	0,786	0,77
media	0,71	0,84	0,76

Tabela 13: Resultados obtidos com neurônios do tipo VG-RAM com limiar de retreino igual a 50.

Base	Precision	Recall	F
1	0,95	0,95	0,95
2	1,00	1,00	1,00
3	0,93	0,93	0,93
4	0,96	0,96	0,96
5	0,77	0,91	0,84
6	0,88	0,96	0,921
7	0,66	0,75	0,703
8	0,531	0,855	0,655
9	0,772	0,815	0,792
10	0,826	0,83	0,828
media	0,70	0,84	0,76

Tabela 14: Resultados obtidos com neurônios do tipo VG-RAM com limiar de retreino igual a 55.

Base	Precision	Recall	F
1	0,93	0,93	0,93
2	1,00	1,00	1,00
3	0,94	0,94	0,94
4	0,94	0,94	0,94
5	0,83	0,95	0,89
6	0,94	1,00	0,968

7	0,72	0,79	0,75
8	0,599	0,913	0,723
9	0,804	0,849	0,826
10	0,807	0,829	0,818
media	0,74	0,87	0,80

Tabela 15: Resultados obtidos com neurônios do tipo VG-RAM com limiar de retreino igual a 60.

Base	Precision	Recall	F
1	0,96	0,96	0,96
2	1,00	1,00	1,00
3	0,94	0,94	0,94
4	0,96	0,96	0,96
5	1,00	1,00	1,00
6	0,86	0,95	0,902
7	0,60	0,66	0,63
8	0,546	0,821	0,656
9	0,775	0,821	0,797
10	0,68	0,68	0,68
media	0,69	0,80	0,74

Tabela 16: Resultados obtidos com neurônios do tipo VG-RAM com limiar de retreino igual a 65.

Base	Precision	Recall	F
1	0,97	0,97	0,97
2	1,00	1,00	1,00
3	0,96	0,96	0,96
4	0,97	0,97	0,97
5	0,93	0,96	0,94
6	0,92	0,99	0,954
7	0,67	0,73	0,699
8	0,511	0,819	0,63
9	0,798	0,822	0,81
10	0,751	0,751	0,751
media	0,70	0,82	0,75

Tabela 17: Resultados obtidos com neurônios do tipo VG-RAM com limiar de retreino igual a 70.

Base	Precision	Recall	F
1	0,93	0,93	0,93
2	1,00	1,00	1,00
3	0,79	0,79	0,79
4	0,95	0,95	0,95
5	1,00	1,00	1,00
6	0,90	0,97	0,934
7	0,58	0,63	0,604
8	0,64	0,837	0,725
9	0,62	0,637	0,628
10	0,685	0,685	0,685

media	0,66	0,74	0,69
-------	------	------	------

Tabela 18: Resultados obtidos com neurônios do tipo VG-RAM com limiar de retreino igual a 80.

Base	Precision	Recall	F
1	0,91	0,91	0,91
2	1,00	1,00	1,00
3	0,99	0,99	0,99
4	0,93	0,96	0,95
5	0,99	0,99	0,99
6	0,92	1,00	0,958
7	0,51	0,54	0,525
8	0,702	0,84	0,765
9	0,654	0,656	0,655
10	0,662	0,662	0,662
media	0,68	0,74	0,71

Tabela 19: Resultados obtidos com neurônios do tipo VG-RAM com limiar de retreino igual a 90.

Base	Precision	Recall	F
1	0,53	0,53	0,53
2	1,00	1,00	1,00
3	0,86	0,86	0,86
4	0,93	0,96	0,95
5	0,99	0,99	0,99
6	0,92	1,00	0,954
7	0,58	0,62	0,598
8	0,797	0,808	0,802
9	0,709	0,725	0,717
10	0,592	0,592	0,592
media	0,72	0,74	0,73

Tabela 20: Resultados obtidos com neurônios do tipo VG-RAM com limiar de retreino igual a 100.

Base	Precision	Recall	F
1	0,53	0,53	0,53
2	1,00	1,00	1,00
3	0,79	0,79	0,79
4	0,31	0,31	0,31
5	0,94	0,94	0,94
6	0,92	1,00	0,956
7	0,44	0,46	0,449
8	0,609	0,728	0,663
9	0,613	0,614	0,613
10	0,595	0,595	0,595
media	0,60	0,65	0,62

O gráfico da Figura 36 apresenta os resultados obtidos com neurônios do tipo VG-RAM *Fat-Fast* com os limiares de retreino 20, 30, 35, 40, 45, 50, 55, 60, 65, 70, 80, 90 e 100. Os vértices presentes em cada curva representam o desempenho alcançado (eixo Y) para cada limiar de retreino (eixo X). É possível perceber que há um ganho significativo no desempenho da rede quando o limiar de retreino é igual a 55. Após este ápice, o desempenho da rede volta a cair.

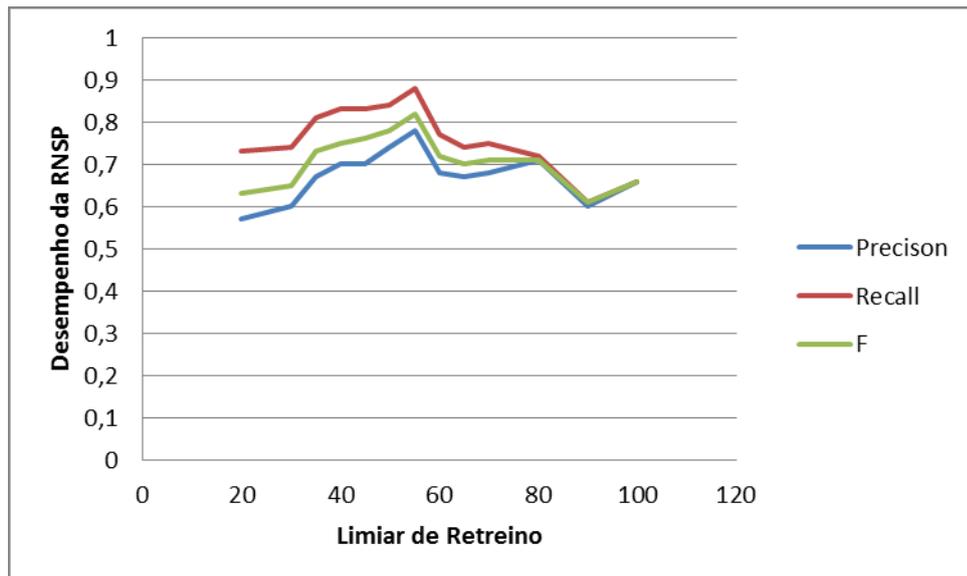


Figura 36: Gráfico que apresenta os resultados do desempenho da RNSP com neurônios do tipo VG-RAM *Fat-Fast* em função dos limiares utilizado para retreino do sistema.

As tabelas 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32 e 33 apresentam numericamente os valores utilizados para representar o gráfico da Figura 36.

Tabela 21: Resultados obtidos com neurônios do tipo VG-RAM *fat-fast* com limiar de retreino igual a 20.

Base	Precision	Recall	F
1	0,91	0,91	0,91
2	1,00	1,00	1,00
3	0,94	0,94	0,94
4	0,29	0,31	0,30
5	0,30	0,33	0,31
6	0,90	0,98	0,939
7	0,58	0,80	0,674
8	0,505	0,832	0,628
9	0,669	0,738	0,702
10	0,194	0,213	0,203
media	0,57	0,73	0,63

Tabela 22: Resultados obtidos com neurônios do tipo VG-RAM *fat-fast* com limiar de retreino igual a 30.

Base	Precision	Recall	F
1	0,95	0,95	0,95
2	1,00	1,00	1,00
3	0,96	0,96	0,96
4	0,92	0,94	0,93
5	0,77	0,83	0,80
6	0,91	0,99	0,945
7	0,58	0,78	0,663
8	0,507	0,83	0,629
9	0,704	0,742	0,722
10	0,228	0,251	0,239
media	0,60	0,74	0,65

Tabela 23: Resultados obtidos com neurônios do tipo VG-RAM *fat-fast* com limiar de retreino igual a 35.

Base	Precision	Recall	F
1	0,77	0,77	0,77
2	1,00	1,00	1,00
3	0,87	0,87	0,87
4	0,95	0,97	0,96
5	0,79	0,86	0,83
6	0,89	0,98	0,931
7	0,62	0,80	0,7
8	0,5	0,826	0,623
9	0,879	0,896	0,887
10	0,349	0,383	0,365
media	0,67	0,81	0,73

Tabela 24: Resultados obtidos com neurônios do tipo VG-RAM *fat-fast* com limiar de retreino igual a 40.

Base	Precision	Recall	F
1	0,96	0,96	0,96
2	1,00	1,00	1,00
3	0,76	0,76	0,76
4	0,97	0,97	0,97
5	0,86	0,97	0,91
6	0,90	0,99	0,941
7	0,74	0,85	0,792
8	0,516	0,852	0,642
9	0,868	0,897	0,882
10	0,392	0,394	0,393
media	0,70	0,83	0,75

Tabela 25: Resultados obtidos com neurônios do tipo VG-RAM *fat-fast* com limiar de retreino igual a 45.

Base	Precision	Recall	F
1	0,97	0,97	0,97
2	1,00	1,00	1,00

3	0,96	0,96	0,96
4	0,97	0,97	0,97
5	0,94	0,97	0,95
6	0,92	1,00	0,954
7	0,76	0,85	0,801
8	0,495	0,819	0,617
9	0,8	0,834	0,817
10	0,719	0,719	0,719
media	0,70	0,83	0,76

Tabela 26: Resultados obtidos com neurônios do tipo VG-RAM *fat-fast* com limiar de retreino igual a 50.

Base	Precision	Recall	F
1	0,95	0,95	0,95
2	1,00	1,00	1,00
3	0,86	0,86	0,86
4	0,97	0,97	0,97
5	1,00	1,00	1,00
6	0,92	1,00	0,953
7	0,73	0,80	0,76
8	0,641	0,903	0,749
9	0,756	0,763	0,76
10	0,813	0,813	0,813
media	0,74	0,84	0,78

Tabela 27: Resultados obtidos com neurônios do tipo VG-RAM *fat-fast* com limiar de retreino igual a 55.

Base	Precision	Recall	F
1	0,91	0,91	0,91
2	1,00	1,00	1,00
3	0,99	0,99	0,99
4	0,97	0,97	0,97
5	1,00	1,00	1,00
6	0,89	0,98	0,933
7	0,77	0,86	0,81
8	0,62	0,893	0,732
9	0,848	0,853	0,85
10	0,853	0,853	0,853
media	0,78	0,88	0,82

Tabela 28: Resultados obtidos com neurônios do tipo VG-RAM *fat-fast* com limiar de retreino igual a 60.

Base	Precision	Recall	F
1	0,90	0,90	0,90
2	1,00	1,00	1,00
3	0,93	0,93	0,93
4	0,97	0,97	0,97
5	0,99	0,99	0,99
6	0,92	1,00	0,955

7	0,66	0,71	0,686
8	0,625	0,89	0,735
9	0,66	0,662	0,661
10	0,67	0,67	0,67
media	0,68	0,77	0,72

Tabela 29: Resultados obtidos com neurônios do tipo VG-RAM *fat-fast* com limiar de retreino igual a 65.

Base	Precision	Recall	F
1	0,96	0,96	0,96
2	1,00	1,00	1,00
3	0,98	0,98	0,98
4	0,97	0,97	0,97
5	0,99	0,99	0,99
6	0,92	0,99	0,951
7	0,65	0,69	0,666
8	0,638	0,826	0,72
9	0,638	0,642	0,64
10	0,645	0,645	0,645
media	0,67	0,74	0,70

Tabela 30: Resultados obtidos com neurônios do tipo VG-RAM *fat-fast* com limiar de retreino igual a 70.

Base	Precision	Recall	F
1	0,96	0,96	0,96
2	1,00	1,00	1,00
3	0,50	0,50	0,50
4	0,97	0,97	0,97
5	0,99	0,99	0,99
6	0,90	0,98	0,935
7	0,72	0,78	0,748
8	0,669	0,841	0,745
9	0,76	0,766	0,763
10	0,207	0,228	0,217
media	0,68	0,75	0,71

Tabela 31: Resultados obtidos com neurônios do tipo VG-RAM *fat-fast* com limiar de retreino igual a 80.

Base	Precision	Recall	F
1	0,92	0,92	0,92
2	1,00	1,00	1,00
3	0,83	0,83	0,83
4	0,84	0,86	0,85
5	0,86	0,97	0,91
6	0,92	1,00	0,956
7	0,74	0,77	0,754
8	0,801	0,809	0,805
9	0,57	0,571	0,57
10	0,699	0,699	0,699

media	0,71	0,72	0,71
-------	------	------	------

Tabela 32: Resultados obtidos com neurônios do tipo VG-RAM *fat-fast* com limiar de retreino igual a 90.

Base	Precision	Recall	F
1	0,92	0,92	0,92
2	1,00	1,00	1,00
3	0,94	0,94	0,94
4	0,39	0,39	0,39
5	0,98	0,99	0,98
6	0,93	0,99	0,962
7	0,63	0,66	0,646
8	0,488	0,489	0,489
9	0,671	0,673	0,672
10	0,445	0,445	0,445
media	0,60	0,61	0,61

Tabela 33: Resultados obtidos com neurônios do tipo VG-RAM *fat-fast* com limiar de retreino igual a 100.

Base	Precision	Recall	F
1	0,83	0,83	0,83
2	1,00	1,00	1,00
3	0,94	0,94	0,94
4	0,93	0,93	0,93
5	0,96	0,98	0,97
6	0,92	1,00	0,958
7	0,51	0,53	0,521
8	0,705	0,706	0,706
9	0,621	0,623	0,622
10	0,532	0,532	0,532
media	0,66	0,66	0,66

5.1.3.2.2 *Variação do Peso dos Neurônios*

Nesta seção apresentamos os resultados do desempenho do sistema de rastreamento visual de objetos de interesse em função do parâmetro s utilizado no cálculo do peso dos neurônios. Para gerar estes resultados utilizamos, estaticamente, o tamanho da memória dos neurônios igual a 32, o limiar de retreino igual a 55 e os demais parâmetros do sistema de rastreamento (o número de *frames* utilizados na janela temporal para escolher a melhor representação atual do objeto no processo de retreino (j) e o número de *pixels* utilizados no ajuste em torno do alvo sacádico do *frame* escolhido na janela temporal com o objetivo de evitar problemas de centralização no processo de retreino (*numPixels*)) iguais a 3.

Conforme apresentado anteriormente, os pesos dos votos dos neurônios w_{ij} são calculados considerando-se a proximidade de dois neurônios ativos em N e a proximidade de seus alvos em Φ . Baseamos esta hipótese no fato de que, se dois neurônios vizinhos, n_{ij} e n_{ij+1} , têm cores de ativação adequadas, ambos devem sinalizar as mesmas coordenadas do centro do objeto de interesse. Expandindo esta hipótese a todos os vizinhos do neurônio n_{ij} e considerando uma janela de vizinhança de raio s (a janela de tamanho $2s+1 \times 2s+1$ pixels), o peso w_{ij} é calculado.

O gráfico da Figura 37 apresenta os resultados obtidos com neurônios do tipo VG-RAM com os valores de s iguais a 1, 2, 3, 4 e 5. Os vértices presentes em cada curva representam o desempenho alcançado (eixo Y) para cada valor de s (eixo X). É possível perceber que há um ganho significativo no desempenho da rede quando s é igual a 4.

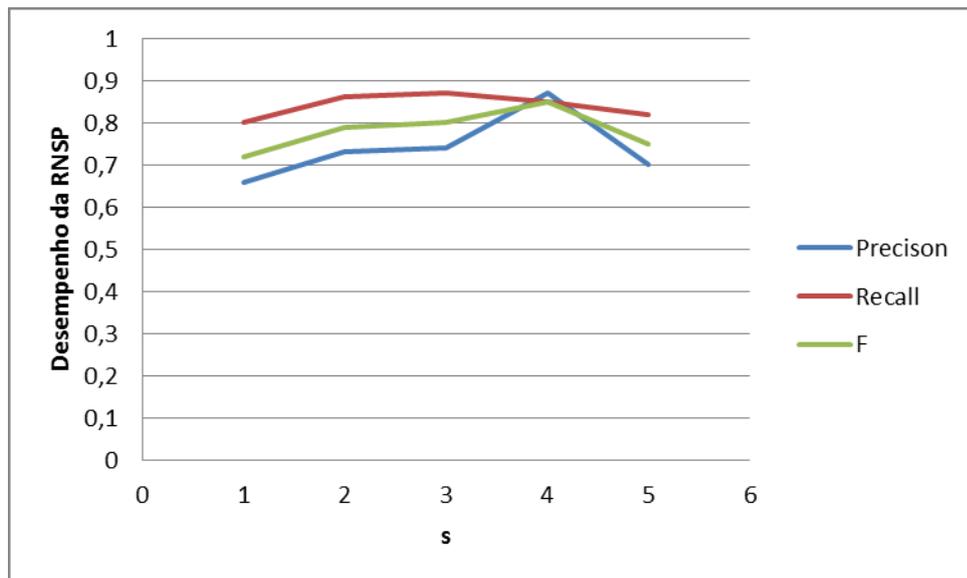


Figura 37: Gráfico que apresenta os resultados do desempenho da RNSP com neurônios do tipo VG-RAM em função de s .

As tabelas 34, 35, 36, 37 e 38 apresentam numericamente os valores utilizados para representar o gráfico da Figura 37.

Tabela 34: Resultados obtidos para neurônios VG-RAM com valor de s igual a 1.

Base	Precision	Recall	F
------	-----------	--------	---

1	0,93	0,93	0,93
2	1,00	1,00	1,00
3	0,84	0,84	0,84
4	0,96	0,96	0,96
5	0,83	0,95	0,89
6	0,71	0,77	0,74
7	0,70	0,79	0,74
8	0,497	0,818	0,618
9	0,711	0,766	0,738
10	0,754	0,775	0,764
Media	0,66	0,80	0,72

Tabela 35: Resultados obtidos para neurônios VG-RAM com valor de s igual a 2.

Base	Precision	Recall	F
1	0,91	0,91	0,91
2	1,00	1,00	1,00
3	0,91	0,91	0,91
4	0,96	0,98	0,97
5	0,83	0,95	0,89
6	0,90	0,98	0,938
7	0,70	0,79	0,745
8	0,509	0,825	0,63
9	0,835	0,887	0,86
10	0,88	0,881	0,88
Media	0,73	0,86	0,79

Tabela 36: Resultados obtidos para neurônios VG-RAM com valor de s igual a 3.

Base	Precision	Recall	F
1	0,93	0,93	0,93
2	1,00	1,00	1,00
3	0,94	0,94	0,94
4	0,94	0,94	0,94
5	0,83	0,95	0,89
6	0,94	1,00	0,968
7	0,72	0,79	0,75
8	0,599	0,913	0,723
9	0,804	0,849	0,826
10	0,807	0,829	0,818
Media	0,74	0,87	0,80

Tabela 37: Resultados obtidos para neurônios VG-RAM com valor de s igual a 4.

Base	Precision	Recall	F
1	1,00	0,94	0,97
2	1,00	1,00	1,00
3	1,00	0,98	0,99
4	1,00	0,95	0,97

5	1,00	0,96	0,98
6	0,92	0,95	0,94
7	0,84	0,97	0,9
8	0,72	0,85	0,78
9	0,93	0,8	0,86
10	1,00	0,8	0,89
Media	0,87	0,85	0,85

Tabela 38: Resultados obtidos para neurônios VG-RAM com valor de s igual a 5.

Base	Precision	Recall	F
1	0,93	0,93	0,93
2	1,00	1,00	1,00
3	0,94	0,94	0,94
4	0,93	0,93	0,93
5	1,00	1,00	1,00
6	0,93	0,97	0,95
7	0,67	0,72	0,692
8	0,553	0,848	0,67
9	0,801	0,841	0,821
10	0,627	0,627	0,627
Media	0,70	0,82	0,75

O gráfico da Figura 38 apresenta os resultados obtidos com neurônios do tipo VG-RAM *Fat-Fast* com os valores de s iguais a 1, 2, 3, 4 e 5. Os vértices presentes em cada curva representam o desempenho alcançado (eixo Y) para cada valor de s (eixo X). É possível perceber que há um ganho significativo no desempenho da rede quando s é igual a 3.

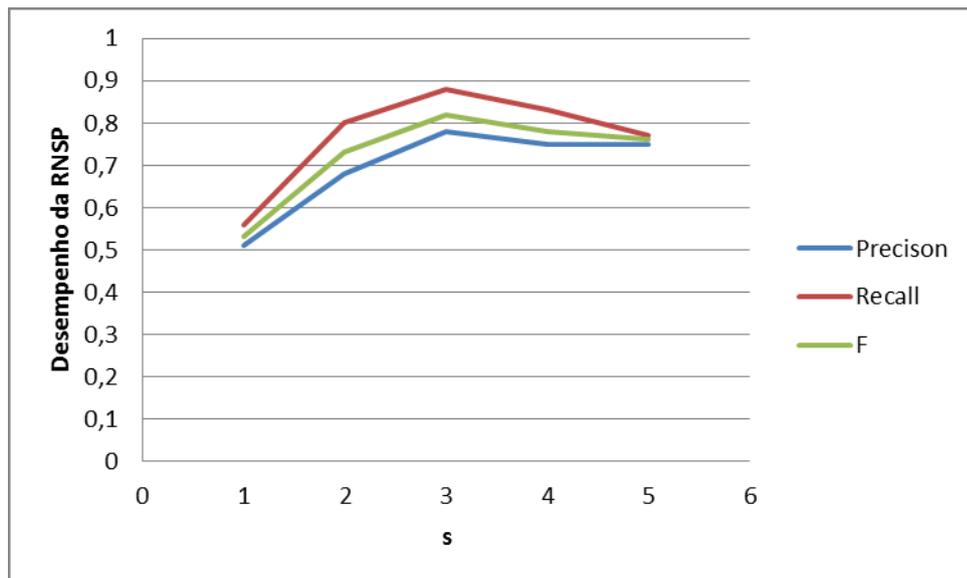


Figura 38: Gráfico que apresenta os resultados do desempenho da RNSP com neurônios do tipo VG-RAM Fat-Fast em função de s.

As tabelas 39, 40, 41, 42 e 43 apresentam numericamente os valores utilizados para representar o gráfico da Figura 38.

Tabela 39: Resultados obtidos para neurônios VG-RAM *fat-fast* com valor de s igual a 1.

Base	Precision	Recall	F
1	0,88	0,88	0,88
2	1,00	1,00	1,00
3	0,92	0,92	0,92
4	0,97	0,97	0,97
5	0,94	0,97	0,96
6	0,92	1,00	0,956
7	0,65	0,74	0,69
8	0,413	0,507	0,455
9	0,535	0,539	0,537
10	0,242	0,266	0,253
Media	0,51	0,56	0,53

Tabela 40: Resultados obtidos para neurônios VG-RAM *fat-fast* com valor de s igual a 2.

Base	Precision	Recall	F
1	0,90	0,90	0,90
2	1,00	1,00	1,00
3	0,98	0,98	0,98
4	0,97	0,97	0,97
5	0,85	1,00	0,92
6	0,88	0,96	0,914
7	0,72	0,81	0,762

8	0,645	0,911	0,755
9	0,712	0,773	0,741
10	0,408	0,414	0,411
Media	0,68	0,80	0,73

Tabela 41: Resultados obtidos para neurônios VG-RAM *fat-fast* com valor de s igual a 3.

Base	Precision	Recall	F
1	0,91	0,91	0,91
2	1,00	1,00	1,00
3	0,99	0,99	0,99
4	0,97	0,97	0,97
5	1,00	1,00	1,00
6	0,89	0,98	0,933
7	0,77	0,86	0,81
8	0,62	0,893	0,732
9	0,848	0,853	0,85
10	0,853	0,853	0,853
Media	0,78	0,88	0,82

Tabela 42: Resultados obtidos para neurônios VG-RAM *fat-fast* com valor de s igual a 4.

Base	Precision	Recall	F
1	0,90	0,90	0,90
2	1,00	1,00	1,00
3	0,80	0,80	0,80
4	0,97	0,97	0,97
5	0,98	0,99	0,98
6	0,92	1,00	0,957
7	0,70	0,75	0,724
8	0,716	0,914	0,803
9	0,812	0,816	0,814
10	0,517	0,568	0,542
Media	0,75	0,83	0,78

Tabela 43: Resultados obtidos para neurônios VG-RAM *fat-fast* com valor de s igual a 5.

Base	Precision	Recall	F
1	0,90	0,90	0,90
2	1,00	1,00	1,00
3	0,63	0,63	0,63
4	0,97	0,97	0,97
5	0,94	0,97	0,95
6	0,89	0,97	0,927
7	0,74	0,80	0,766
8	0,94	0,948	0,944
9	0,742	0,746	0,744
10	0,134	0,147	0,14
Media	0,75	0,77	0,76

5.1.3.2.3 *Variação do Número de Frames Utilizados na Janela Temporal*

Nesta seção apresentamos os resultados do desempenho do sistema de rastreamento visual de objetos de interesse em função do número de *frames* utilizados na janela temporal utilizada no processo de retreino. Para gerar estes resultados utilizamos, estaticamente, o tamanho da memória dos neurônios igual a 32, o limiar de retreino igual a 55 e os demais parâmetros do sistema de rastreamento (o parâmetro s utilizado no cálculo do peso dos neurônios e o número de *pixels* utilizados no ajuste em torno do alvo sacádico do *frame* escolhido na janela temporal com o objetivo de evitar problemas de centralização no processo de retreino (*numPixels*)) iguais a 3.

Conforme apresentado anteriormente, é necessário garantir uma boa escolha da aparência do objeto a ser utilizada no retreino do sistema. Para tal, inicialmente, consideramos uma janela temporal (últimos j *frames*) para escolher a melhor representação atual do objeto.

O gráfico da Figura 39 apresenta os resultados obtidos com neurônios do tipo VG-RAM para os valores de j iguais a 1, 2, 3, 4 e 5. Os vértices presentes em cada curva representam o desempenho alcançado (eixo Y) para cada valor de j (eixo X). É possível perceber que há um ganho significativo no desempenho da rede quando o valor de j é 3.

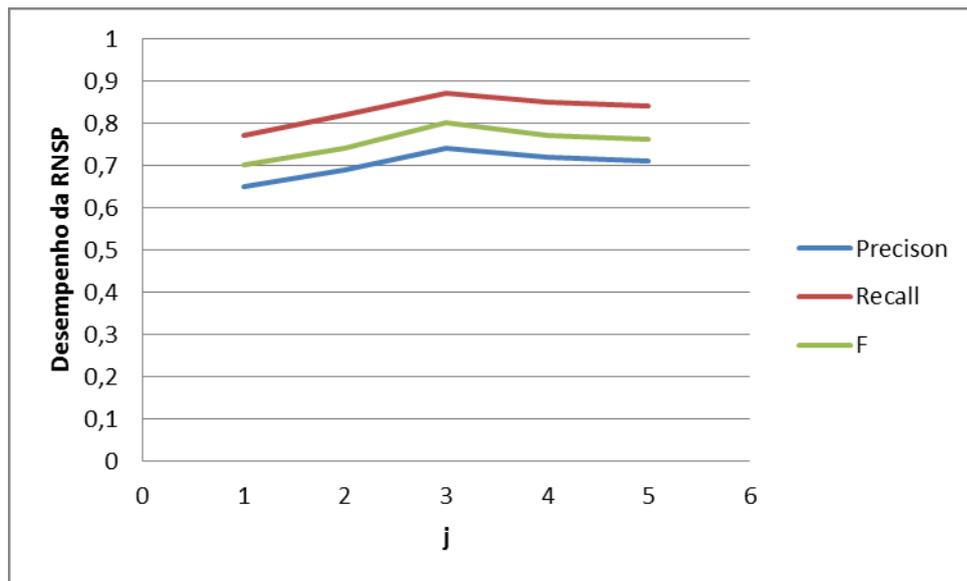


Figura 39: Gráfico que apresenta os resultados do desempenho da RNSP com neurônios do tipo VG-RAM em função de j .

As tabelas 44, 45, 46, 47 e 48 apresentam numericamente os valores utilizados para representar o gráfico da Figura 39.

Tabela 44: Resultados obtidos para neurônios VG-RAM com valor de j igual a 1.

Base	Precision	Recall	F
1	0,97	0,97	0,97
2	1,00	1,00	1,00
3	0,77	0,77	0,77
4	0,95	0,96	0,95
5	1,00	1,00	1,00
6	0,89	0,95	0,917
7	0,70	0,78	0,734
8	0,534	0,836	0,652
9	0,663	0,701	0,682
10	0,644	0,659	0,652
Media	0,65	0,77	0,70

Tabela 45: Resultados obtidos para neurônios VG-RAM com valor de j igual a 2.

Base	Precision	Recall	F
1	0,94	0,94	0,94
2	1,00	1,00	1,00
3	0,80	0,80	0,80
4	0,94	0,94	0,94
5	0,83	0,95	0,89
6	0,92	1,00	0,959
7	0,70	0,77	0,729

8	0,521	0,846	0,645
9	0,759	0,81	0,784
10	0,734	0,734	0,734
Media	0,69	0,82	0,74

Tabela 46: Resultados obtidos para neurônios VG-RAM com valor de j igual a 3.

Base	Precision	Recall	F
1	0,93	0,93	0,93
2	1,00	1,00	1,00
3	0,94	0,94	0,94
4	0,94	0,94	0,94
5	0,83	0,95	0,89
6	0,94	1,00	0,968
7	0,72	0,79	0,75
8	0,599	0,913	0,723
9	0,804	0,849	0,826
10	0,807	0,829	0,818
Media	0,74	0,87	0,80

Tabela 47: Resultados obtidos para neurônios VG-RAM com valor de j igual a 4.

Base	Precision	Recall	F
1	0,93	0,93	0,93
2	1,00	1,00	1,00
3	0,79	0,79	0,79
4	0,97	0,97	0,97
5	0,84	0,96	0,89
6	0,91	0,99	0,951
7	0,62	0,70	0,655
8	0,53	0,844	0,651
9	0,84	0,885	0,862
10	0,756	0,756	0,756
Media	0,72	0,85	0,77

Tabela 48: Resultados obtidos para neurônios VG-RAM com valor de j igual a 5.

Base	Precision	Recall	F
1	0,92	0,92	0,92
2	1,00	1,00	1,00
3	0,78	0,78	0,78
4	0,96	0,96	0,96
5	1,00	1,00	1,00
6	0,88	0,96	0,916
7	0,71	0,77	0,739
8	0,525	0,842	0,647
9	0,784	0,831	0,807
10	0,771	0,79	0,781

Media	0,71	0,84	0,76
-------	------	------	------

O gráfico da Figura 40 apresenta os resultados obtidos com neurônios do tipo VG-RAM *Fat-Fast* para os valores de j iguais a 1, 2, 3, 4 e 5. Os vértices presentes em cada curva representam o desempenho alcançado (eixo Y) para cada valor de j (eixo X). É possível perceber que há um ganho significativo no desempenho da rede quando o valor de j é 3.

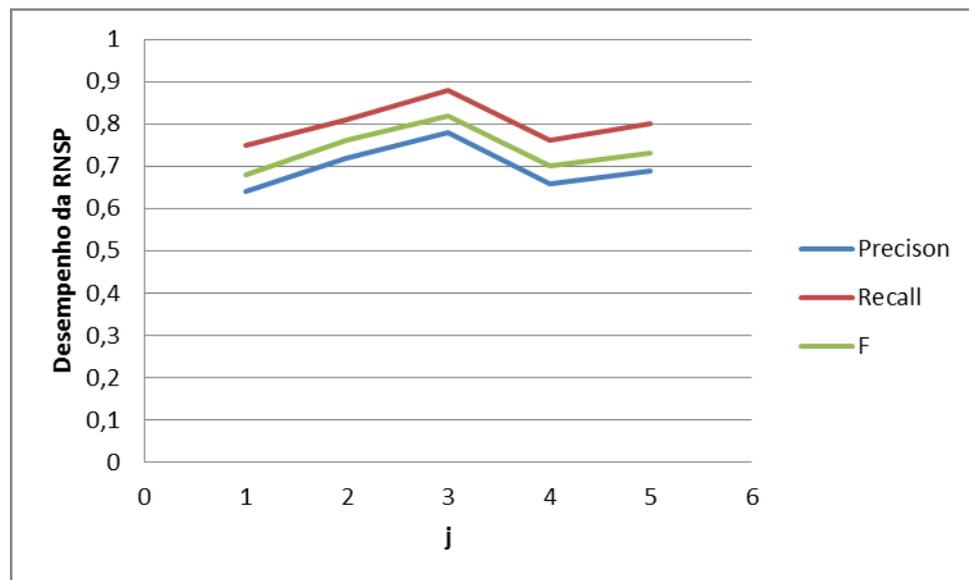


Figura 40: Gráfico que apresenta os resultados do desempenho da RNSP com neurônios do tipo VG-RAM *Fat-Fast* em função de j .

As tabelas 49, 50, 51, 52 e 53 apresentam numericamente os valores utilizados para representar o gráfico da Figura 40.

Tabela 49: Resultados obtidos para neurônios VG-RAM *fat-fast* com valor de j igual a 1.

Base	Precision	Recall	F
1	0,91	0,91	0,91
2	1,00	1,00	1,00
3	0,95	0,95	0,95
4	0,97	0,97	0,97
5	0,77	0,84	0,80
6	0,90	0,98	0,935
7	0,74	0,81	0,776
8	0,533	0,825	0,648
9	0,691	0,701	0,696
10	0,441	0,441	0,441
Media	0,64	0,75	0,68

Tabela 50: Resultados obtidos para neurônios VG-RAM *fat-fast* com valor de j igual a 2.

Base	Precision	Recall	F
1	0,95	0,95	0,95
2	1,00	1,00	1,00
3	0,86	0,86	0,86
4	0,97	0,97	0,97
5	0,94	0,97	0,96
6	0,90	0,99	0,942
7	0,68	0,75	0,71
8	0,622	0,873	0,727
9	0,854	0,861	0,858
10	0,41	0,411	0,41
Media	0,72	0,81	0,76

Tabela 51: Resultados obtidos para neurônios VG-RAM *fat-fast* com valor de j igual a 3.

Base	Precision	Recall	F
1	0,91	0,91	0,91
2	1,00	1,00	1,00
3	0,99	0,99	0,99
4	0,97	0,97	0,97
5	1,00	1,00	1,00
6	0,89	0,98	0,933
7	0,77	0,86	0,81
8	0,62	0,893	0,732
9	0,848	0,853	0,85
10	0,853	0,853	0,853
Media	0,78	0,88	0,82

Tabela 52: Resultados obtidos para neurônios VG-RAM *fat-fast* com valor de j igual a 4.

Base	Precision	Recall	F
1	0,96	0,96	0,96
2	1,00	1,00	1,00
3	0,89	0,89	0,89
4	0,97	0,97	0,97
5	1,00	1,00	1,00
6	0,92	1,00	0,958
7	0,75	0,81	0,777
8	0,584	0,841	0,689
9	0,727	0,739	0,733
10	0,352	0,363	0,357
Media	0,66	0,76	0,70

Tabela 53: Resultados obtidos para neurônios VG-RAM *fat-fast* com valor de j igual a 5.

Base	Precision	Recall	F
1	0,96	0,96	0,96

2	1,00	1,00	1,00
3	0,94	0,94	0,94
4	0,97	0,97	0,97
5	1,00	1,00	1,00
6	0,88	0,95	0,915
7	0,78	0,87	0,823
8	0,539	0,835	0,655
9	0,714	0,718	0,716
10	0,742	0,742	0,742
Media	0,69	0,80	0,73

5.1.3.2.4 *Variação do Número de Pixels Utilizados no Ajuste do Alvo Sacádico*

Nesta seção apresentamos os resultados do desempenho do sistema de rastreamento visual de objetos de interesse em função do número de *pixels* utilizados no ajuste do alvo sacádico no processo de retreino. Para gerar estes resultados utilizamos, estaticamente, o tamanho da memória dos neurônios igual a 32, o limiar de retreino igual a 55 e os demais parâmetros do sistema de rastreamento (o parâmetro s utilizado no cálculo do peso dos neurônios e o número de *frames* utilizados na janela temporal para escolher a melhor representação atual do objeto no processo de retreino (j)) iguais a 3.

Conforme apresentado anteriormente, é necessário garantir uma boa escolha da aparência do objeto a ser utilizada no retreino do sistema. Para tal, inicialmente, consideramos uma janela temporal para escolher a melhor representação atual do objeto. Em seguida, realizamos um ajuste em torno do alvo sacádico do *frame* escolhido na janela temporal com o objetivo de evitar problemas de centralização. Neste ajuste, o centro de atenção é movido para cada um dos *pixels* de uma janela de $numPixels \times numPixels$ *pixels* e a matriz acumuladora é recalculada para cada *pixel*. O *pixel* com o valor mais alto de confiança é escolhido como o centro do objeto de interesse que será utilizado para retreinar o sistema.

O gráfico da Figura 41 apresenta os resultados obtidos com neurônios do tipo VG-RAM para os valores de *numPixels* iguais a 1, 2, 3, 4 e 5. Os vértices presentes em cada curva representam o desempenho alcançado (eixo Y) para cada valor de *numPixels* (eixo X). É possível perceber que há um ganho significativo para os valores 3 e 5.

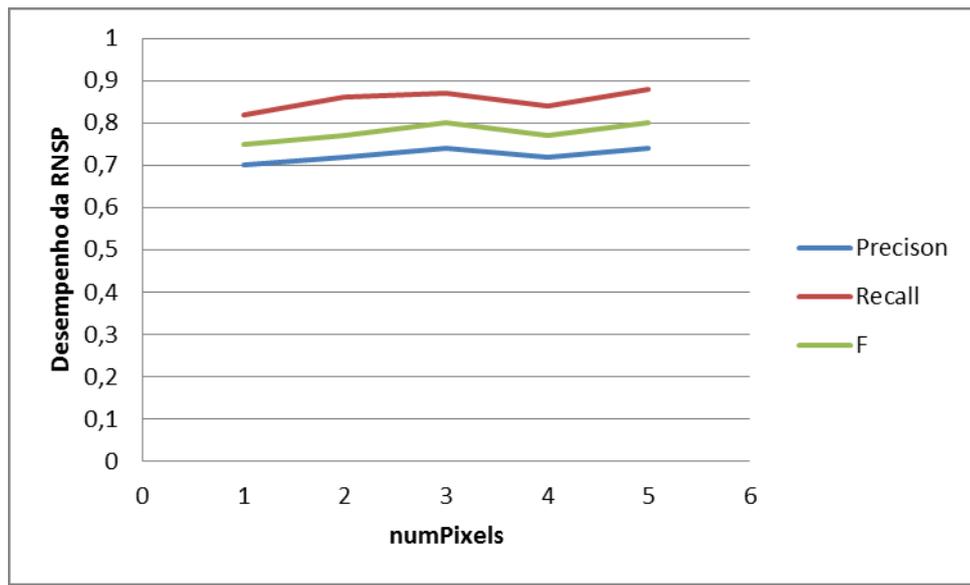


Figura 41: Gráfico que apresenta os resultados do desempenho da RNSP com neurônios do tipo VG-RAM em função de numPixels.

As tabelas 54, 55, 56, 57 e 58 apresentam numericamente os valores utilizados para representar o gráfico da Figura 41.

Tabela 54: Resultados obtidos para neurônios VG-RAM com valor de numPixels igual a 1.

Base	Precision	Recall	F
1	0,96	0,96	0,96
2	1,00	1,00	1,00
3	0,98	0,98	0,98
4	0,96	0,96	0,96
5	0,82	0,94	0,88
6	0,92	1,00	0,956
7	0,64	0,71	0,67
8	0,562	0,871	0,683
9	0,749	0,791	0,769
10	0,777	0,777	0,777
Media	0,70	0,82	0,75

Tabela 55: Resultados obtidos para neurônios VG-RAM com valor de numPixels igual a 2.

Base	Precision	Recall	F
1	0,95	0,95	0,95
2	1,00	1,00	1,00
3	0,92	0,92	0,92
4	0,96	0,96	0,96
5	0,96	0,97	0,97
6	0,89	0,97	0,931
7	0,66	0,74	0,698

8	0,557	0,894	0,686
9	0,798	0,847	0,822
10	0,775	0,781	0,778
media	0,72	0,86	0,77

Tabela 56: Resultados obtidos para neurônios VG-RAM com valor de *numPixels* igual a 3.

Base	Precision	Recall	F
1	0,93	0,93	0,93
2	1,00	1,00	1,00
3	0,94	0,94	0,94
4	0,94	0,94	0,94
5	0,83	0,95	0,89
6	0,94	1,00	0,968
7	0,72	0,79	0,75
8	0,599	0,913	0,723
9	0,804	0,849	0,826
10	0,807	0,829	0,818
Media	0,74	0,87	0,80

Tabela 57: Resultados obtidos para neurônios VG-RAM com valor de *numPixels* igual a 4.

Base	Precision	Recall	F
1	0,96	0,96	0,96
2	1,00	1,00	1,00
3	0,94	0,94	0,94
4	0,98	0,98	0,98
5	0,88	0,96	0,92
6	0,88	0,96	0,915
7	0,73	0,81	0,767
8	0,537	0,831	0,652
9	0,805	0,854	0,829
10	0,723	0,753	0,738
Media	0,72	0,84	0,77

Tabela 58: Resultados obtidos para neurônios VG-RAM com valor de *numPixels* igual a 5.

Base	Precision	Recall	F
1	0,95	0,95	0,95
2	1,00	1,00	1,00
3	0,90	0,90	0,90
4	0,94	0,94	0,94
5	0,84	0,95	0,89
6	0,87	0,95	0,911
7	0,73	0,79	0,761
8	0,573	0,926	0,708
9	0,831	0,881	0,856
10	0,747	0,765	0,756
Media	0,74	0,88	0,80

O gráfico da Figura 42 apresenta os resultados obtidos com neurônios do tipo VG-RAM *Fat-Fast* para os valores de *numPixels* iguais a 1, 2, 3, 4 e 5. Os vértices presentes em cada curva representam o desempenho alcançado (eixo Y) para cada valor de *numPixels* (eixo X). É possível perceber que há um ganho significativo no desempenho da rede quando *numPixels* é igual a 3.

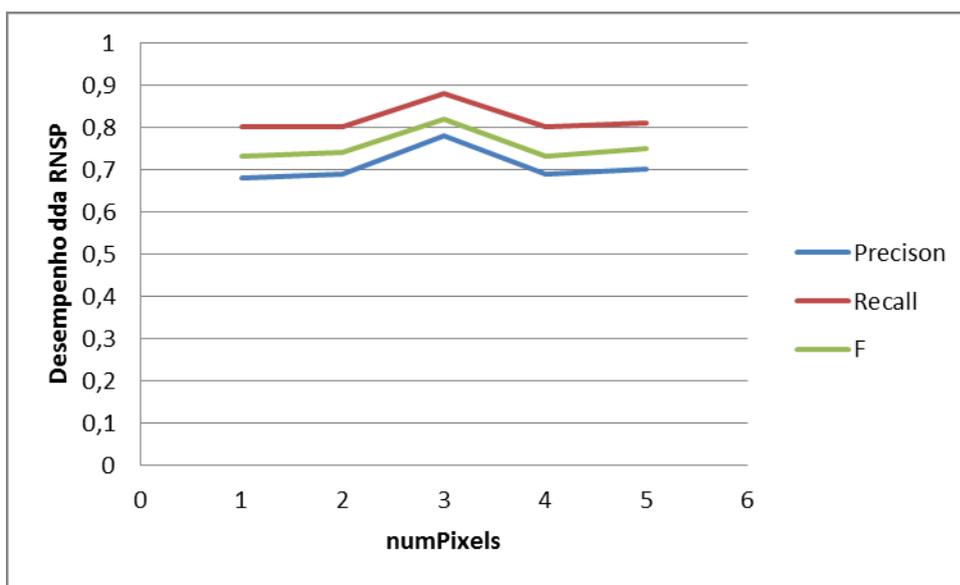


Figura 42: Gráfico que apresenta os resultados do desempenho da RNSP com neurônios do tipo VG-RAM *Fat-Fast* em função de *numPixels*.

As tabelas 59, 60, 61, 62 e 63 apresentam numericamente os valores utilizados para representar o gráfico da Figura 42.

Tabela 59: Resultados obtidos para neurônios VG-RAM *fat-fast* com valor de *numPixels* igual a 1.

Base	Precision	Recall	F
1	0,92	0,92	0,92
2	1,00	1,00	1,00
3	0,94	0,94	0,94
4	0,93	0,95	0,94
5	0,98	0,99	0,98
6	0,88	0,97	0,923
7	0,76	0,85	0,799
8	0,555	0,835	0,667
9	0,797	0,835	0,816
10	0,378	0,385	0,382
Media	0,68	0,80	0,73

Tabela 60: Resultados obtidos para neurônios VG-RAM *fat-fast* com valor de *numPixels* igual a 2.

Base	Precision	Recall	F
1	0,88	0,88	0,88
2	1,00	1,00	1,00
3	0,99	0,99	0,99
4	0,97	0,97	0,97
5	0,94	0,97	0,95
6	0,91	1,00	0,953
7	0,73	0,81	0,768
8	0,676	0,954	0,791
9	0,741	0,745	0,743
10	0,322	0,354	0,337
Media	0,69	0,80	0,74

Tabela 61: Resultados obtidos para neurônios VG-RAM *fat-fast* com valor de *numPixels* igual a 3.

Base	Precision	Recall	F
1	0,91	0,91	0,91
2	1,00	1,00	1,00
3	0,99	0,99	0,99
4	0,97	0,97	0,97
5	1,00	1,00	1,00
6	0,89	0,98	0,933
7	0,77	0,86	0,81
8	0,62	0,893	0,732
9	0,848	0,853	0,85
10	0,853	0,853	0,853
Media	0,78	0,88	0,82

Tabela 62: Resultados obtidos para neurônios VG-RAM *fat-fast* com valor de *numPixels* igual a 4.

Base	Precision	Recall	F
1	0,96	0,96	0,96
2	1,00	1,00	1,00
3	0,83	0,83	0,83
4	0,97	0,97	0,97
5	0,95	0,97	0,96
6	0,86	0,95	0,902
7	0,60	0,66	0,629
8	0,527	0,839	0,647
9	0,831	0,849	0,84
10	0,514	0,514	0,514
Media	0,69	0,80	0,73

Tabela 63: Resultados obtidos para neurônios VG-RAM *fat-fast* com valor de *numPixels* igual a 5.

Base	Precision	Recall	F
1	0,88	0,88	0,88
2	1,00	1,00	1,00

3	0,94	0,94	0,94
4	0,97	0,97	0,97
5	0,95	0,97	0,96
6	0,91	0,98	0,941
7	0,75	0,83	0,787
8	0,546	0,823	0,656
9	0,767	0,78	0,774
10	0,718	0,718	0,718
Media	0,70	0,81	0,75

5.1.3.2.5 Variação da Memória dos Neurônios

Nesta seção apresentamos os resultados do desempenho do sistema de rastreamento visual de objetos de interesse em função da quantidade de memória dos neurônios. Para gerar estes resultados utilizamos, estaticamente, o limiar de retreino igual a 55 e os demais parâmetros do sistema de rastreamento (o parâmetro s utilizado no cálculo do peso dos neurônios, o número de *frames* utilizados na janela temporal para escolher a melhor representação atual do objeto no processo de retreino (j), o número de *pixels* utilizados no ajuste em torno do alvo sacádico do *frame* escolhido na janela temporal com o objetivo de evitar problemas de centralização no processo de retreino (*numPixels*)) iguais a 3.

Conforme apresentado anteriormente, no modelo proposto foi utilizada uma camada neural de 65 x 48 neurônios e uma tabela-verdade (memória) com tamanho de $65 \times 48 \times m$. Esta quantidade de entradas é equivalente a m linhas por neurônio na tabela-verdade, ou seja, o sistema tem memória para armazenar m representações completas do objeto de interesse.

O gráfico da Figura 43 apresenta os resultados obtidos com neurônios do tipo VG-RAM para os valores de m iguais a 1, 4, 8, 16, 32, 64 e 128. Os vértices presentes em cada curva representam o desempenho alcançado (eixo Y) para cada valor de m (eixo X). É possível perceber que foi alcançado um ótimo desempenho com valor de m igual a 32.

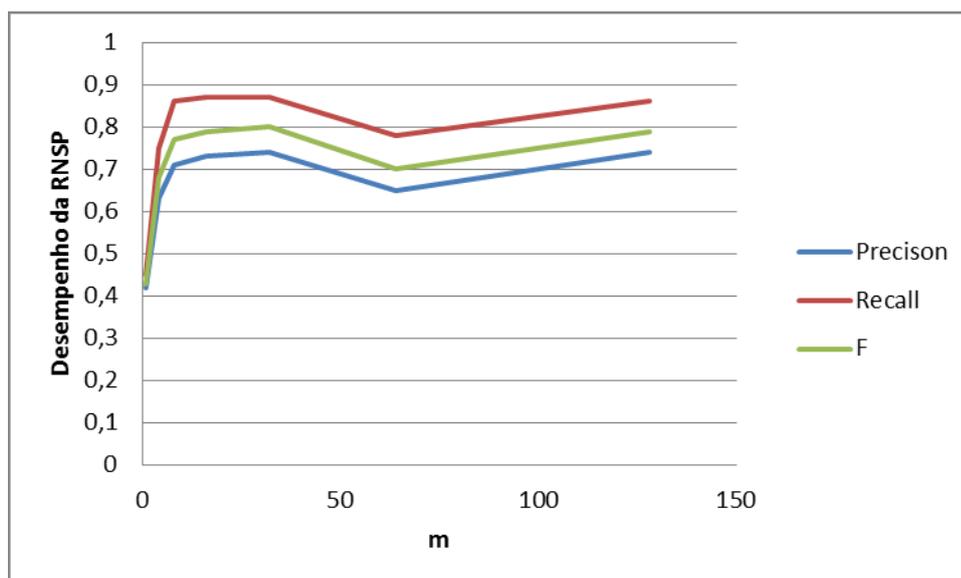


Figura 43: Gráfico que apresenta os resultados do desempenho da RNSP com neurônios do tipo VG-RAM em função de m .

As tabelas 64, 65, 66, 67, 68, 69 e 70 apresentam numericamente os valores utilizados para representar o gráfico da Figura 43.

Tabela 64: Resultados obtidos para neurônios VG-RAM com valor de m igual a 1.

Base	Precision	Recall	F
1	0,65	0,65	0,65
2	1,00	1,00	1,00
3	0,89	0,89	0,89
4	0,96	0,96	0,96
5	0,85	0,98	0,91
6	0,92	0,98	0,949
7	0,31	0,35	0,324
8	0,054	0,089	0,067
9	0,576	0,615	0,595
10	0,627	0,648	0,637
Media	0,42	0,45	0,43

Tabela 65: Resultados obtidos para neurônios VG-RAM com valor de m igual a 4.

Base	Precision	Recall	F
1	0,91	0,91	0,91
2	1,00	1,00	1,00
3	0,79	0,79	0,79
4	0,96	0,96	0,96
5	0,83	0,95	0,89
6	0,86	0,94	0,9
7	0,67	0,74	0,701

8	0,531	0,829	0,648
9	0,696	0,751	0,723
10	0,39	0,39	0,39
Media	0,63	0,75	0,68

Tabela 66: Resultados obtidos para neurônios VG-RAM com valor de m igual a 8.

Base	Precision	Recall	F
1	0,97	0,97	0,97
2	1,00	1,00	1,00
3	0,86	0,86	0,86
4	0,96	0,96	0,96
5	0,83	0,95	0,89
6	0,90	0,98	0,94
7	0,64	0,70	0,665
8	0,555	0,915	0,691
9	0,853	0,903	0,877
10	0,577	0,577	0,577
Media	0,71	0,86	0,77

Tabela 67: Resultados obtidos para neurônios VG-RAM com valor de m igual a 16.

Base	Precision	Recall	F
1	0,96	0,96	0,96
2	1,00	1,00	1,00
3	0,83	0,83	0,83
4	0,96	0,96	0,96
5	0,83	0,95	0,89
6	0,90	0,98	0,942
7	0,68	0,79	0,733
8	0,514	0,829	0,635
9	0,848	0,897	0,872
10	0,84	0,84	0,84
Media	0,73	0,87	0,79

Tabela 68: Resultados obtidos para neurônios VG-RAM com valor de m igual a 32.

Base	Precision	Recall	F
1	0,93	0,93	0,93
2	1,00	1,00	1,00
3	0,94	0,94	0,94
4	0,94	0,94	0,94
5	0,83	0,95	0,89
6	0,94	1,00	0,968
7	0,72	0,79	0,75
8	0,599	0,913	0,723
9	0,804	0,849	0,826
10	0,807	0,829	0,818

Media	0,74	0,87	0,80
-------	------	------	------

Tabela 69: Resultados obtidos para neurônios VG-RAM com valor de m igual a 64.

Base	Precision	Recall	F
1	0,96	0,96	0,96
2	1,00	1,00	1,00
3	0,86	0,86	0,86
4	0,97	0,97	0,97
5	0,83	0,95	0,89
6	0,90	0,98	0,941
7	0,73	0,79	0,761
8	0,549	0,855	0,669
9	0,623	0,663	0,642
10	0,739	0,755	0,747
Media	0,65	0,78	0,70

Tabela 70: Resultados obtidos para neurônios VG-RAM com valor de m igual a 128.

Base	Precision	Recall	F
1	0,61	0,61	0,61
2	1,00	1,00	1,00
3	0,89	0,89	0,89
4	0,96	0,96	0,96
5	0,83	0,95	0,89
6	0,87	0,96	0,913
7	0,69	0,76	0,722
8	0,619	0,901	0,734
9	0,818	0,86	0,839
10	0,824	0,824	0,824
Media	0,74	0,86	0,79

O gráfico da Figura 44 apresenta os resultados obtidos com neurônios do tipo VG-RAM *Fat-Fast* para os valores de m iguais a 1, 4, 8, 16, 32, 64 e 128. Os vértices presentes em cada curva representam o desempenho alcançado (eixo Y) para cada valor de m (eixo X). É possível perceber que o melhor desempenho foi alcançado com valor de m iguais a 8 e 32.

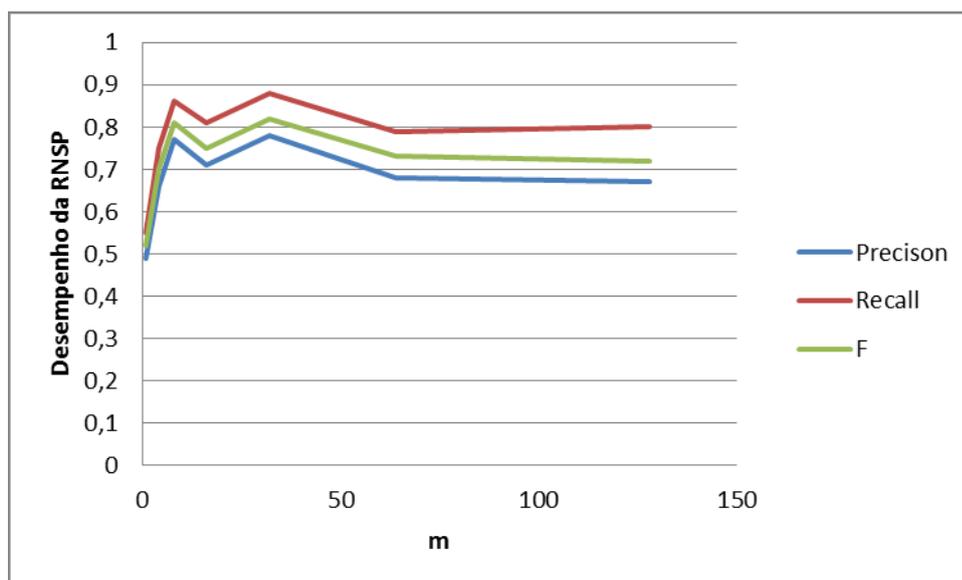


Figura 44: Gráfico que apresenta os resultados do desempenho da RNSP com neurônios do tipo VG-RAM Fat-Fast em função de m .

As tabelas 71, 72, 73, 74, 75, 76 e 77 apresentam numericamente os valores utilizados para representar o gráfico da Figura 44.

Tabela 71: Resultados obtidos para neurônios VG-RAM *fat-fast* com valor de m igual a 1.

Base	Precision	Recall	F
1	0,96	0,96	0,96
2	1,00	1,00	1,00
3	0,87	0,87	0,87
4	0,97	0,97	0,97
5	0,93	0,96	0,94
6	0,94	0,99	0,965
7	0,39	0,42	0,401
8	0,715	0,838	0,772
9	0,295	0,322	0,308
10	0,17	0,187	0,178
Media	0,49	0,55	0,52

Tabela 72: Resultados obtidos para neurônios VG-RAM *fat-fast* com valor de m igual a 4.

Base	Precision	Recall	F
1	0,96	0,96	0,96
2	1,00	1,00	1,00
3	0,60	0,60	0,60
4	0,97	0,97	0,97
5	0,95	0,97	0,96
6	0,90	0,98	0,939
7	0,66	0,70	0,679

8	0,623	0,83	0,712
9	0,739	0,777	0,758
10	0,31	0,31	0,31
Media	0,66	0,75	0,70

Tabela 73: Resultados obtidos para neurônios VG-RAM *fat-fast* com valor de m igual a 8.

Base	Precision	Recall	F
1	0,96	0,96	0,96
2	1,00	1,00	1,00
3	0,99	0,99	0,99
4	0,97	0,97	0,97
5	0,59	0,61	0,60
6	0,90	0,99	0,942
7	0,72	0,78	0,75
8	0,617	0,866	0,721
9	0,879	0,888	0,883
10	0,781	0,781	0,781
Media	0,77	0,86	0,81

Tabela 74: Resultados obtidos para neurônios VG-RAM *fat-fast* com valor de m igual a 16.

Base	Precision	Recall	F
1	0,96	0,96	0,96
2	1,00	1,00	1,00
3	0,69	0,69	0,69
4	0,97	0,97	0,97
5	0,93	0,96	0,95
6	0,91	0,99	0,948
7	0,69	0,73	0,707
8	0,534	0,811	0,644
9	0,834	0,844	0,838
10	0,645	0,645	0,645
Media	0,71	0,81	0,75

Tabela 75: Resultados obtidos para neurônios VG-RAM *fat-fast* com valor de m igual a 32.

Base	Precision	Recall	F
1	0,91	0,91	0,91
2	1,00	1,00	1,00
3	0,99	0,99	0,99
4	0,97	0,97	0,97
5	1,00	1,00	1,00
6	0,89	0,98	0,933
7	0,77	0,86	0,81
8	0,62	0,893	0,732
9	0,848	0,853	0,85
10	0,853	0,853	0,853

Media	0,78	0,88	0,82
-------	------	------	------

Tabela 76: Resultados obtidos para neurônios VG-RAM *fat-fast* com valor de m igual a 64.

Base	Precision	Recall	F
1	0,90	0,90	0,90
2	1,00	1,00	1,00
3	0,70	0,70	0,70
4	0,97	0,97	0,97
5	0,98	0,99	0,98
6	0,89	0,97	0,931
7	0,70	0,78	0,739
8	0,621	0,911	0,738
9	0,67	0,68	0,675
10	0,686	0,689	0,687
Media	0,68	0,79	0,73

Tabela 77: Resultados obtidos para neurônios VG-RAM *fat-fast* com valor de m igual a 128.

Base	Precision	Recall	F
1	0,89	0,89	0,89
2	1,00	1,00	1,00
3	0,86	0,86	0,86
4	0,97	0,97	0,97
5	0,94	0,97	0,95
6	0,92	1,00	0,956
7	0,71	0,83	0,76
8	0,53	0,853	0,654
9	0,845	0,878	0,861
10	0,23	0,253	0,241
Media	0,67	0,80	0,72

5.1.4 Resultados

Nesta seção apresentamos os melhores resultados obtidos no rastreamento de objetos de interesse para as imagens do conjunto de dados do TLD.

Os melhores resultados obtidos no rastreamento de objetos de interesse para as imagens do conjunto de dados do TLD utilizando neurônios VG-RAM foram alcançados segundo esta configuração de parâmetros:

- Número de neurônios da rede igual a 65×48 ;

- Número de sinapses por neurônio igual a 256;
- Tamanho da entrada de rede igual a 201×201 ;
- Desvio padrão igual a 10;
- Fator de log igual a 2;
- Limiar de retreino (Ω) igual a 55;
- Parâmetro s utilizado no cálculo do peso dos neurônios igual a 4;
- Número de *frames* utilizados na janela temporal para escolher a melhor representação atual do objeto no processo de retreino (j) igual a 3;
- Número de *pixels* utilizados no ajuste em torno do alvo sacádico do *frame* escolhido na janela temporal com o objetivo de evitar problemas de centralização no processo de retreino (*numPixels*) igual a 3;
- Tamanho da memória dos neurônios igual a 32.

A Tabela 78 apresenta os resultados obtidos com neurônios do tipo VG-RAM e um vídeo com os resultados alcançados está disponível em <https://www.youtube.com/watch?v=rz5-5lG6yU>.

Tabela 78: Resultados do TLD e obtidos com o sistema de rastreamento visual utilizando neurônios VG-RAM.

Video	Frames	TLD			Sistema de Rastreamento Visual		
		Precision	Recall	F	Precision	Recall	F
David (1)	761	1,00	1,00	1	1,00	0,94	0,97
Jumping (2)	313	1,00	1,00	1	1,00	1,00	1,00
Pedestrian1 (3)	140	1,00	1,00	1	1,00	0,98	0,99
Pedestrian2 (4)	338	0,89	0,92	0,91	1,00	0,95	0,97
Pedestrian3 (5)	184	0,99	1,00	0,99	1,00	0,96	0,98
Car (6)	945	0,92	0,97	0,94	0,92	0,95	0,94
Motocross (7)	2665	0,89	0,77	0,83	0,84	0,97	0,90
Volkswagen (8)	8576	0,80	0,96	0,87	0,72	0,85	0,78
Carchase (9)	9928	0,86	0,70	0,77	0,93	0,80	0,86
Panda (10)	3000	0,58	0,63	0,60	1,00	0,80	0,89
Media		0,82	0,81	0,81	0,86	0,85	0,85

Os coeficientes de jaccard obtidos nestes experimentos são apresentados na Figura 45 como *boxplots* dos coeficientes de jaccard obtidos nos *frames* válidos (ou seja, *frames* com objeto presente). O *boxplot* utiliza caixas para representar o valor médio com um desvio padrão para baixo e uma para cima e linhas para representar o valor máximo e mínimo alcançado em cada vídeo. Podemos perceber que as médias dos coeficientes de jaccard obtidas pelo sistema de rastreamento visual são valores bem elevados, representando uma grande acurácia do sistema.

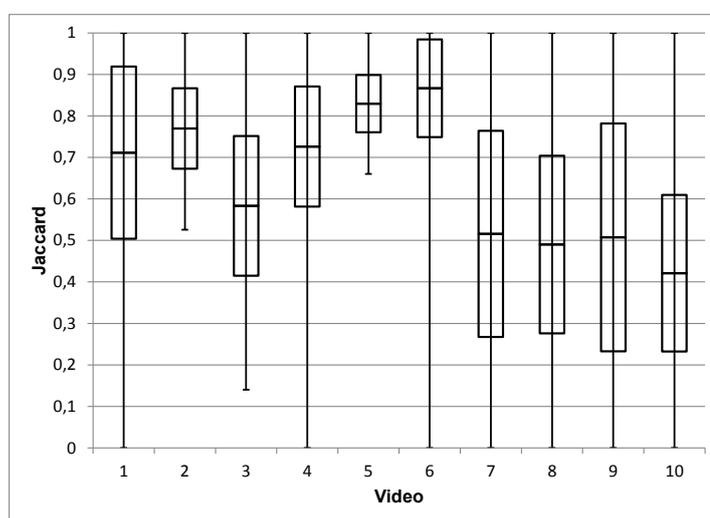


Figura 45: Resumo dos coeficientes de jaccard por frame para todos os vídeos. As caixas representam o valor médio com um desvio padrão para baixo e um para cima e as linhas representam os valores mínimo e máximo alcançados.

Além da avaliação quantitativa, apresentamos uma análise qualitativa dos resultados obtidos com o sistema de rastreamento visual utilizando neurônios VG-

RAM para cada vídeo testado. A seguir a saída do sistema é apresentada como um retângulo verde e o *ground-truth* anotado manualmente é apresentado como um retângulo vermelho. Para permitir a análise dos resultados, os vídeos foram amostrados regularmente.

Os resultados do rastreamento visual dos *frames* 1, 190, 380, 570, e 761 do vídeo David são apresentados na Figura 46. Como podemos observar, a variação de iluminação não impedem o rastreador visual identificar corretamente o objeto de interesse (face) na cena. Como mostrado na Tabela 78, as mudanças na aparência do objeto resultou em alguns falsos negativos.



Figura 46: Resultados do rastreamento visual dos frames selecionados do vídeo David (da esquerda para a direita, Frame 1, Frame 190, Frame 380, Frame 570 e Frame 761). Figura retirada de [61].

A Figura 47 apresenta os resultados do rastreamento visual dos *frames* 1, 78, 156, 234 e 313 do vídeo Jumping. Como podemos ver, o movimento da câmera e os borrões na imagem não impedem o sistema de rastreamento identificar corretamente o objeto de interesse (face). Para este vídeo o sistema atingiu a máxima performance possível.



Figura 47: Resultados do rastreamento visual dos frames selecionados do vídeo Jumping (da esquerda para a direita, Frame 1, Frame 78, Frame 156, Frame 234 e Frame 313). Figura retirada de [61].

O resultado do rastreamento visual dos *frames* 1, 35, 70, 105 e 140 do vídeo Pedestrian1 são mostrados na Figura 48. Neste vídeo há variações abruptas da aparência do objeto de interesse, afetando o Recall do sistema. O número de falsos negativos aumenta uma vez que o módulo (Re)Learning não é capaz de atualizar as representações do objeto na memória dos neurônios rapidamente (há um atraso de

10 *frames* entre retreinos). Podemos notar, ainda, que o *ground-truth* pode estar impactando negativamente nos resultados. No *frame* 70, por exemplo, a caixa delimitadora do *ground-truth* é muito maior do que a resposta dada pelo sistema. No entanto, podemos perceber que a caixa delimitadora resultante do sistema circunscreve o objeto de interesse (a pessoa) muito melhor do que a caixa delimitadora do *ground-truth*.



Figura 48: Resultados do rastreamento visual dos frames selecionados do vídeo Pedestrian1 (da esquerda para a direita, Frame 1, Frame 35, Frame 70, Frame 105 e Frame 140). Figura retirada de [61].

A Figura 49 mostra os resultados do sistema para os *frames* 1, 84, 169, 253, e 338 do vídeo Pedestrian2. A Figura 50 mostra os resultados para os *frames* 1, 46, 92, 138 e 184 do vídeo Pedestrian3. A Figura 51 apresenta o resultado para os *frames* 1, 236, 472, 708 e 945 do vídeo Car. O sistema realiza satisfatoriamente o rastreamento para todos esses casos.

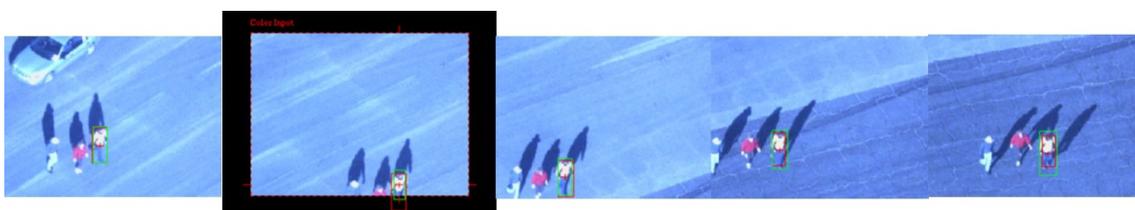


Figura 49: Resultados do rastreamento visual dos frames selecionados do vídeo Pedestrian2 (da esquerda para a direita, Frame 1, Frame 84, Frame 169, Frame 253 e Frame 338). Figura retirada de [61].

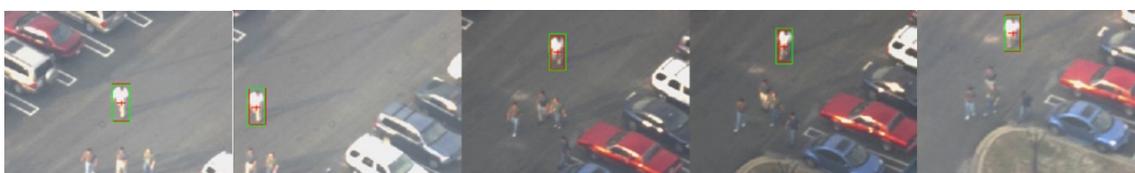


Figura 50: Resultados do rastreamento visual dos frames selecionados do vídeo Pedestrian3 (da esquerda para a direita, Frame 1, Frame 46, Frame 92, Frame 138 e Frame 184). Figura retirada de [61].



Figura 51: Resultados do rastreamento visual dos frames seleccionados do vídeo Car (da esquerda para a direita, Frame 1, Frame 236, Frame 472, Frame 708 e Frame 945). Figura retirada de [61].

O resultado do rastreamento visual dos *frames* 1, 666, 1332, 1998, and 2665 do vídeo Motocross é apresentado na Figura 52. Falsos Positivos ocorrem devido às mudanças abruptas de escala, pois o sistema estima corretamente o centro do objeto, mas não estima a escala corretamente (a interseção das caixas delimitadoras não é suficiente para apontar uma detecção correta do objeto). É interessante notar, ainda, que o sistema detecta corretamente o objeto de interesse no *frame* 1998, mas o *ground-truth* não aponta objeto de interesse presente neste *frame*. O *frame* 2665 apresenta um exemplo de Falso Positivo.



Figura 52: Resultados do rastreamento visual dos frames seleccionados do vídeo Motocross (da esquerda para a direita, Frame 1, Frame 666, Frame 1332, Frame 1998 e Frame 2665). Figura retirada de [61].

A Figura 53 apresenta o resultado do sistema para os *frames* 1, 2144, 4288, 6432, e 8576 do vídeo Volkswagen. A performance do sistema não é tão boa pois neste vídeo o módulo Detection identifica outros objetos com formas bem similares como sendo o objeto de interesse. Um exemplo desta situação pode ser visto no *frame* 6632, no qual um Falso Positivo ocorre.



Figura 53: Resultados do rastreamento visual dos frames seleccionados do vídeo Volkswagen (da esquerda para a direita, Frame 1, Frame 2144, Frame 4288, Frame 6432 e Frame 8576). Figura retirada de [61].

A Figura 54 mostra o resultado do rastreamento visual para os *frames* 1, 2482, 4964, 7446, and 9928 do vídeo Carchase e a Figura 55 apresenta o resultado dos *frames* 1, 750, 1500, 2250, and 3000 do vídeo Panda. Como pode ser notado nas imagens, o sistema rastreia os objetos de interesse durante todo o vídeo.



Figura 54: Resultados do rastreamento visual dos frames selecionados do vídeo Carchase (da esquerda para a direita, Frame 1, Frame 2482, Frame 4964, Frame 7446 e Frame 9928). Figura retirada de [61].



Figura 55: Resultados do rastreamento visual dos frames selecionados do vídeo Panda (da esquerda para a direita, Frame 1, Frame 750, Frame 1500, Frame 2250 e Frame 3000). Figura retirada de [61].

No geral, os resultados apresentados anteriormente mostram que o sistema proposto de rastreamento visual biologicamente inspirado é capaz de obter resultados similares ou superiores aos obtidos pelas técnicas estado-da-arte para rastreamento de objetos em vídeos. O sistema proposto é capaz de rastrear objetos em vídeos longos que possuem muitos desafios, como oclusões totais ou parciais, movimentos abruptos de câmeras, mudanças drásticas na aparência do objeto de interesse, etc.

O desempenho do sistema de rastreamento visual de objetos proposto também pode ser medido considerando o tempo de processamento. Como dito anteriormente, executamos os experimentos em uma estação de trabalho com processador Intel Core i7-4770 *quad-core* de 3,4 GHz e 16 GB de memória RAM. Utilizando um neurônio VG-RAM, o treino inicial do sistema pode ser realizado em 0,01 segundos e o retreino do sistema pode ser realizado em 0,32 segundos, em média. O tempo de retreino é mais elevado dado que uma busca “fina” (considerando uma janela de pesquisa de 3×3 *pixels*) é realizada em torno do alvo sacádico a fim de assegurar a centralização do objeto a ser treinado. Nesta busca, o centro de atenção é transferido para cada um dos *pixels* da janela de busca e a matriz de acumulação é recalculada. Já o procedimento de rastreamento (módulo Tracking) compreende três etapas que podem ser executadas em 0,34 segundos, em média: a primeira sequência de sacadas, realizada em 0,12 segundos, em média; o ajuste de escala, realizado em 0,1 segundos, em média; e a segunda

sacada, também realizada em 0,12 segundos, em média. Por fim, a detecção de um objeto é realizada utilizando cerca de 20 escalas vezes 4 sacadas por *frame*, exigindo 80 sacadas de 0,12 segundos. Esta abordagem exaustiva foi escolhida para assegurar que o objeto será encontrado no primeiro *frame* possível. No entanto, estamos trabalhando em uma pesquisa com base randômica que pode otimizar este processo, quando encontrar o objeto no primeiro *frame* possível não seja obrigatório.

O número total de ativações do módulo Detection e o número de *frames* utilizados para retreino em cada vídeo são mostrados na Tabela 79. Percebemos, ao analisar o número total de ativações do módulo Detection, que este é, em geral, não muito superior ao número de *frames* inválidos dos vídeos (*frames* em que o objeto de interesse não estava presente na cena e, portanto, a detecção é necessária), exceto para o vídeos 1, 6 e 9. No vídeo 1, o objeto de interesse gira em torno de seu eixo muito rapidamente, não permitindo ao sistema aprender a nova aparência do objeto de interesse com certeza elevada, alterando o estado do sistema para "o objeto não é visível" levando à ativação do módulo Detection. No vídeo 6, o sistema realiza o rastreamento corretamente do objeto em *frames* onde o *ground-truth* diz não haver objeto e, portanto, ativando o módulo Detection menos vezes do que o esperado considerando o número de *frames* inválidos. Por fim, no vídeo 9 o objeto sofre oclusão e reaparece inúmeras vezes com aparência e escala muito diferente. A fim de assegurar a detecção do objeto corretamente para estes casos, o sistema exige um elevado limiar de confiança para detecção que, por conseguinte, aumenta o número de vezes que o módulo Detection é ativado. Percebemos, ainda, ao olhar para o número de *frames* utilizados para o retreino que este é, em geral, muito menor do que o número de *frames* válidos. Tal fato nos permite inferir que os neurônios podem memorizar a aparência de um objeto ao longo das sequências dos vídeos com poucas amostras do objeto.

Tabela 79: Número de retreinos realizados e número de *frames* em que o módulo Detection foi ativado.

Video	Número de Frames	Número de Frames Inválidos	Número de Retreinos	Número de Frames em que o módulo Detection foi ativado
David (1)	761	0	18	46
Jumping (2)	313	0	16	0
Pedestrian1 (3)	140	0	9	0
Pedestrian2 (4)	338	72	10	80
Pedestrian3 (5)	184	28	7	29
Car (6)	945	85	11	46
Motocross (7)	2665	1253	88	1337
Volkswagen (8)	8576	3435	84	3519
Carchase (9)	9928	1268	218	2347
Panda (10)	3000	270	96	366

A otimização apresentada do neurônio VG-RAM - o neurônio VG-RAM *Fat-Fast* – tem gerado um ganho considerável de tempo de execução do sistema de rastreamento visual proposto e com resultados bem relevantes. O tempo de execução do sistema utilizando o neurônio VG-RAM *Fat-Fast* tem sido em média 1/3 do tempo de execução do sistema utilizando o neurônio VG-RAM. Portanto, o sistema executa todo o processo requerido para o rastreamento do objeto, em média, em 0,1 segundo utilizando neurônio VG-RAM *Fat-Fast*. É importante mencionar, ainda, que a nossa implementação pode ser otimizada para tirar vantagem de aceleradores de *hardware*, como a GPU, FPGA ou processadores de sinais digitais, melhorando os indicadores de desempenho de tempo que acabamos de apresentar.

Os melhores resultados obtidos no rastreamento de objetos de interesse para as imagens do conjunto de dados do TLD utilizando neurônios VG-RAM *Fat-Fast* foram alcançados segundo esta configuração de parâmetros:

- Número de neurônios da rede igual a 65×48 ;
- Número de sinapses por neurônio igual a 256;
- Tamanho da entrada de rede igual a 201×201 ;
- Desvio padrão igual a 10;
- Fator de log igual a 2;

- Limiar de retreino (Ω) igual a 55;
- Parâmetro s utilizado no cálculo do peso dos neurônios igual a 3;
- Número de *frames* utilizados na janela temporal para escolher a melhor representação atual do objeto no processo de retreino (j) igual a 3;
- Número de *pixels* utilizados no ajuste em torno do alvo sacádico do *frame* escolhido na janela temporal com o objetivo de evitar problemas de centralização no processo de retreino (*numPixels*) igual a 3;
- Tamanho da memória dos neurônios igual a 32.

A Tabela 80 apresenta os resultados obtidos com neurônios do tipo VG-RAM *Fat-Fast*. Podemos perceber, nesta tabela, que os resultados obtidos com os neurônios VG-RAM *Fat-Fast* são pouco inferiores em relação aos obtidos com neurônios VG-RAM (compare com a Tabela 78).

Tabela 80: Resultados obtidos com o sistema de rastreamento visual utilizando neurônios VG-RAM *Fat-Fast*.

Base	Precision	Recall	F
1	0,91	0,91	0,91
2	1,00	1,00	1,00
3	0,99	0,99	0,99
4	0,97	0,97	0,97
5	1,00	1,00	1,00
6	0,89	0,98	0,933
7	0,77	0,86	0,81
8	0,62	0,893	0,732
9	0,848	0,853	0,85
10	0,853	0,853	0,853
Media	0,78	0,88	0,82

5.2 Experimento Siga-o-Líder

Um dos objetivos do Laboratório de Computação de Alto Desempenho (LCAD) do Departamento de Informática (DI) da Universidade Federal do Espírito Santo

(UFES) é a navegação autônoma da IARA em dois percursos: a Volta da UFES e a Ida a Guarapari.

Na Volta da UFES o objetivo é realizar uma volta completa ao redor do campus de Goiabeiras da UFES. Este campus é o principal da UFES e possui um anel viário com 3.570 metros (Figura 56). Já na Ida a Guarapari o objetivo é tornar a IARA capaz de realizar uma viagem partindo da UFES com destino à cidade de Guarapari autonomamente. O percurso total da UFES à cidade de Guarapari possui 58,5 Km.

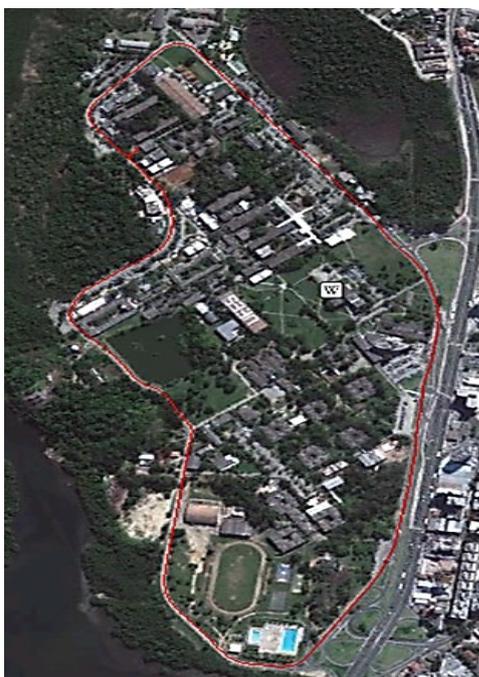


Figura 56: Anel viário da UFES.

Um das formas possíveis de realizar tais objetivos (ou auxiliar para que estes sejam alcançados) é fazer com que a IARA siga um carro-líder que lhe guiará em seus percursos (siga-o-líder), tanto na Volta da UFES quanto na Ida a Guarapari.

Nesta seção apresentamos a metodologia e o resultado obtido no experimento “Siga-o-líder”.

5.2.1 Metodologia

Nesta seção apresentamos a metodologia utilizada na implementação da arquitetura proposta neste experimento.

5.2.1.1 A Plataforma Robótica IARA

Para testar os algoritmos apresentados neste experimento, utilizamos a plataforma robótica IARA (*Intelligent Autonomous Robotic Automobile*), o carro autônomo do LCAD (<http://www.lcad.inf.ufes.br>). IARA é um automóvel de passeio Ford Escape Hybrid (Figura 57) adaptado com sensores e com a possibilidade de acionamento dos atuadores (volante, acelerador, freio, entre outros) por meio de computadores instalados no porta-malas (Figura 58). A tecnologia eletrônica de acionamento dos atuadores do automóvel foi desenvolvida pela empresa *Torc Robotics* [68].



Figura 57: Plataforma IARA.



Figura 58: Recursos computacionais da plataforma IARA.

Na plataforma robótica IARA estão instalados os seguintes sensores: câmeras estéreo *Bumblebee XB3* da *Point Grey* (Colúmbia Britânica, Canadá) (<http://ww2.ptgrey.com/stereo-vision/bumblebee-xb3>), *Light Detection And Ranging* (LiDAR) HDL-32E da *Velodyne* (Califórnia, Estados Unidos) (<http://velodynelidar.com/lidar/hdlproducts/hdl32e.aspx>) e o *Attitude and Heading Reference System* (AHRS) MTi da *Xsens* (Califórnia, Estados Unidos) (<http://www.xsens.com/products/mti/>). A Figura 59 ilustra de como os sensores foram instalados na plataforma robótica IARA.



Figura 59: Sensores instalados na IARA.

Para processar todos os dados dos sensores, IARA também conta com seis computadores Dell Precision R5500 (2 Processadores Intel Xeon 2.13 GHZ, 12 GB de memória DDR3 1333MHZ, 2 HDs SSD 120GB em RAID0, 2 Placas de Rede 1GB, Placa de Vídeo Quadro 600, Placa de Vídeo Tesla C2050). Todos os computadores utilizaram sistema operacional Ubuntu 12.04 com *kernel real-time*. O uso de sistemas operacionais com *kernel real-time* foi importante para garantir uma alta precisão nas medidas de tempo dos sensores.

5.2.1.2 O *Framework* CARMEN

Os algoritmos apresentados neste experimento foram desenvolvidos na linguagem C/C++ como um módulo na plataforma CARMEN toolkit. Carmen Robot Navigation Toolkit ou CARMEN [69] é uma coleção de software aberto GPL (<http://carmen.sourceforge.net>), desenvolvido na Carnegie Mellon University (CMU), para controle de robôs. O *framework* utiliza uma arquitetura orientada a serviços (SOA - *Service Oriented Architecture*) e tem como objetivos diminuir a barreira entre o desenvolvimento de novos códigos tanto para robôs simulados quanto para robôs reais e facilitar o compartilhamento de algoritmos entre diferentes instituições [70].

Um módulo desenvolvido utilizando o *framework* CARMEN deve ser isolado de todos os outros programas do sistema, comunicando-se apenas através de mensagens. Isso gera um baixo acoplamento entre os módulos e minimiza interferências no funcionamento de outros módulos, aumentando a robustez do sistema como um todo.

A comunicação entre os módulos é feita através de um protocolo de comunicação chamado de *Inter Process Communication* (IPC) [71], utilizando principalmente o paradigma de comunicação *Publish-Subscribe*. A implementação deste paradigma disponibilizada pelo IPC possui diversas vantagens como: flexibilidade, eficiência e a capacidade de trocar mensagens com estruturas de dados complexas, incluindo listas e vetores de tamanho variável. Além disso, um atrativo do IPC é a possibilidade de comunicação entre processos sendo executados em diferentes computadores por meio do protocolo TCP/IP.

5.2.2 Implementação

Na implementação do modelo matemático-computacional do Superior Colliculus utilizamos o framework MAE com neurônios do tipo VG-RAM e empregamos a plataforma robótica CARMEN para a integração do sistema de rastreamento visual de objetos com o sistema de câmeras estéreo e de coleta de dados. Inicialmente, deve ser anotada manualmente a caixa delimitadora do objeto de interesse. Após tal demarcação, o sistema de rastreamento visual será treinado e funcionará conforme apresentado no capítulo anterior.

Os principais componentes deste sistema são:

- Central: o módulo é responsável por manter as informações, rotear e fazer o registro do tráfego de mensagens do sistema.
- Bumblebee: o módulo `bumblebee_basic` foi desenvolvido pelo grupo do LCAD e tem a tarefa de capturar imagens retificadas de câmeras Bumblebee e publicá-las para os demais módulos.
- Rastreamento Visual: o módulo é uma aplicação do *framework* MAE que implementa a arquitetura do modelo de Rastreamento Visual descrito no capítulo anterior. Esta aplicação também é um módulo CARMEN, pois se conecta ao processo central e assina mensagens do módulo `bumblebee_basic`. O módulo escuta mensagens `bumblebee_basic` nos *buffers* de comunicação do CARMEN, manda executar o rastreamento visual caso haja um novo frame (uma nova mensagem `bumblebee_basic`) e, ao final de um rastreamento, publica mensagens com a posição do objeto na imagem e a confiança desta informação ou com a informação que o objeto não está presente na cena. Caso o objeto de interesse sofra oclusão, nos *frames* seguintes, um centro de atenção inicial aleatório é escolhido e uma sequência de movimentos sacádicos é realizada. Nesta implementação, com o objetivo de ser mais rápida a execução do módulo Detection, não é realizada a exploração total da cena.

5.2.3 Resultado

Um primeiro teste com o IARA seguindo um carro-líder foi realizado em torno do anel viário da UFES e pode ser visto no vídeo disponível em <https://www.youtube.com/watch?v=eQY38H44RDw>. Podemos perceber, neste vídeo, que o IARA segue o objeto de interesse (carro-líder) durante todo o percurso.

Como o resultado do módulo de rastreamento visual de objetos é a posição do objeto na imagem e a confiança de esta informação ser verdadeira, é possível que o sistema saiba quando o objeto está ou quando o objeto não está sendo rastreado. No vídeo deste primeiro teste, por exemplo, o sistema perde o objeto durante um pequeno percurso (fazendo com que o IARA navegue de forma autônoma) e o detecta em seguida, permitindo novamente que o IARA siga o carro-líder até o final do percurso.

Um segundo teste com o IARA seguindo um carro-líder em torno do anel viário da UFES foi realizado e pode ser visto no vídeo disponível em <https://www.youtube.com/watch?v=lePu4KskvNk>. Podemos perceber, neste vídeo, que o IARA também segue o objeto de interesse (carro-líder) durante todo o percurso.

Neste vídeo, inicialmente, é anotada manualmente a caixa delimitadora do objeto de interesse no sistema desenvolvido. A Figura 60 apresenta a janela utilizada para demarcação da caixa delimitadora do objeto de interesse a ser rastreado. Nesta janela, clica-se nos limites superior esquerdo e inferior direito (ou vice-versa) do objeto de interesse para definir a caixa delimitadora do objeto a ser rastreado. Nesta figura, o cursor está posicionado sobre o limite inferior direito do carro-líder a ser seguido.

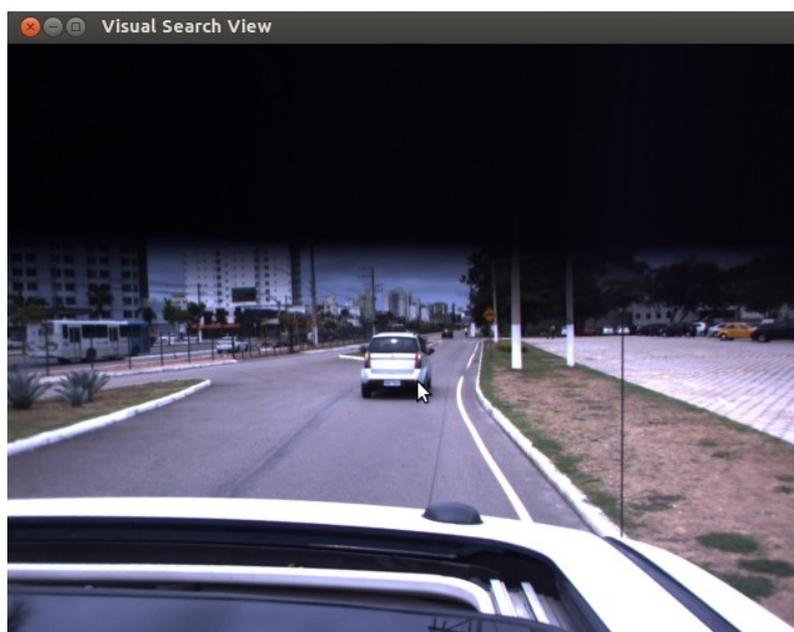


Figura 60: Janela utilizada para demarcação da caixa delimitadora do objeto de interesse a ser rastreado.

Após tal demarcação, o sistema de rastreamento visual é treinado e funciona conforme apresentado no capítulo anterior. A Figura 61 apresenta o resultado (uma cruz vermelha no centro do objeto) do Sistema de Rastreamento Visual para um *frame* subsequente ao treinado.

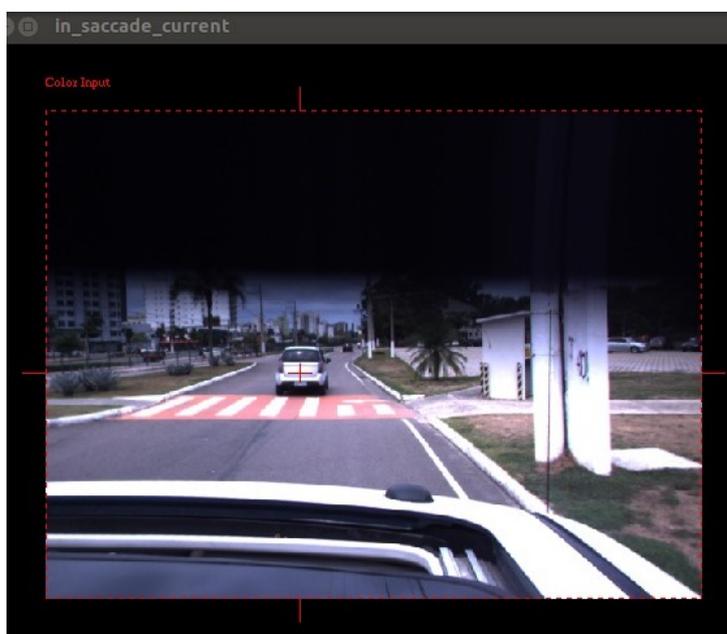


Figura 61: Resultado do Sistema de Rastreamento Visual para um frame subsequente ao treinado.

Por fim, o ponto resultado do Sistema de Rastreamento Visual é mapeado para o mapa 2D utilizado para navegação do IARA. Este ponto será o próximo objetivo a ser alcançado pelo IARA. A Figura 62 apresenta o mapa 2D utilizado. O retângulo branco é a representação do IARA e o retângulo amarelo é a representação em 2D do objetivo a ser alcançado obtido através do Sistema, ou seja, da posição em 2D do objeto que esta sendo rastreado no mapa de navegação.

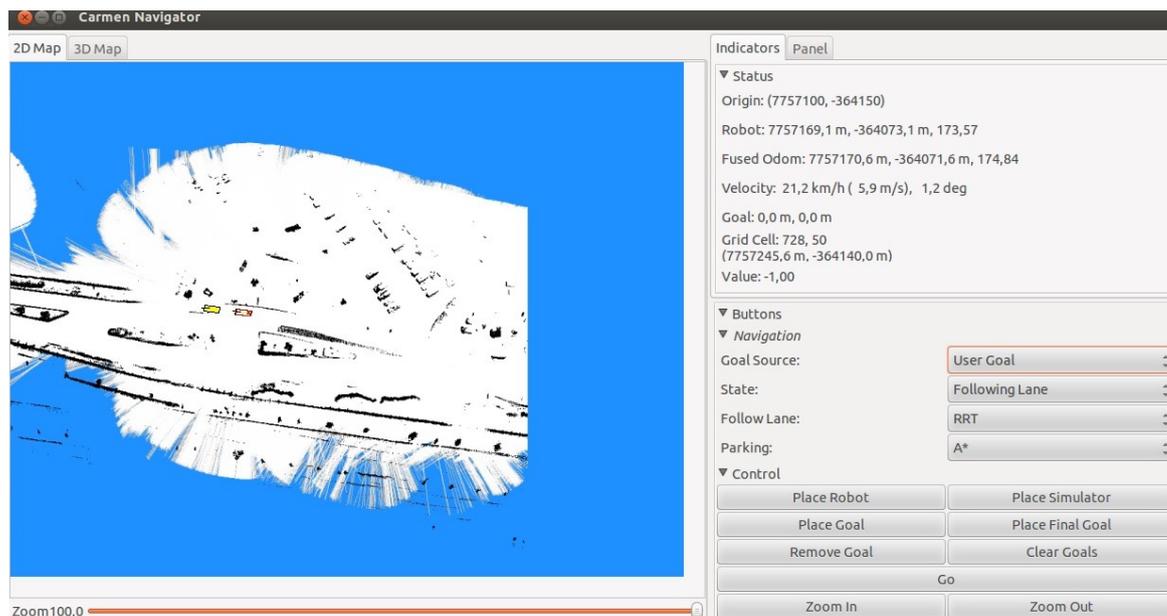


Figura 62: Mapa de navegação da IARA. O retângulo amarelo é o próximo objetivo a ser alcançado pelo IARA. O retângulo branco é o IARA representado no mapa 2D.

Tais resultados serão importantes no desenvolvimento das estratégias que serão adotadas para que o IARA complete a ida a Guarapari.

5.3 Experimento com o *Eye-tracker*

Um experimento com um *eye-tracker* foi realizado com o objetivo de comparar de uma forma preliminar o sistema proposto com o sistema visual humano. Um *eye-tracker* permite medir e registrar os movimentos oculares de um indivíduo a partir da amostragem de um estímulo em um ambiente real ou controlado, determinando,

deste modo, em que áreas tal indivíduo fixa a sua atenção e em que ordem segue em sua exploração visual.

Os resultados das fixações dos olhos, dos movimentos oculares (sacadas), da dilatação da pupila e do piscar de olhos podem ser analisados com o intuito de demonstrar evidências de padrões visuais específicos. Para tal, contamos com o auxílio de um *software* (disponibilizado com o óculos *eye-tracker*) capaz de criar representações que resumem graficamente o comportamento visual de um utilizador de um óculos *eye-tracker*, demonstrando em uma imagem a forma como o usuário explora a interface.

Utilizamos, em nosso experimento, um óculos *eye-tracker* SMI (<http://www.smivision.com/en/gaze-and-eye-tracking-systems/home.html>). A Figura 63 apresenta o óculos *eye-tracker* utilizado.



Figura 63: Eye-tracker SMI.

Um vídeo com o experimento realizado e com um comparativo entre os resultados obtidos com o *eye-tracker* e com o Sistema de Rastreamento Visual proposto esta disponível em <https://www.youtube.com/watch?v=MIMlxHp4uz0>. Neste vídeo, o resultado das fixações obtidas com o *eye-tracker* são representadas por círculos vermelhos na interface (Figura 64) e os resultados obtidos com o sistema são representados por retângulos (Figura 65).

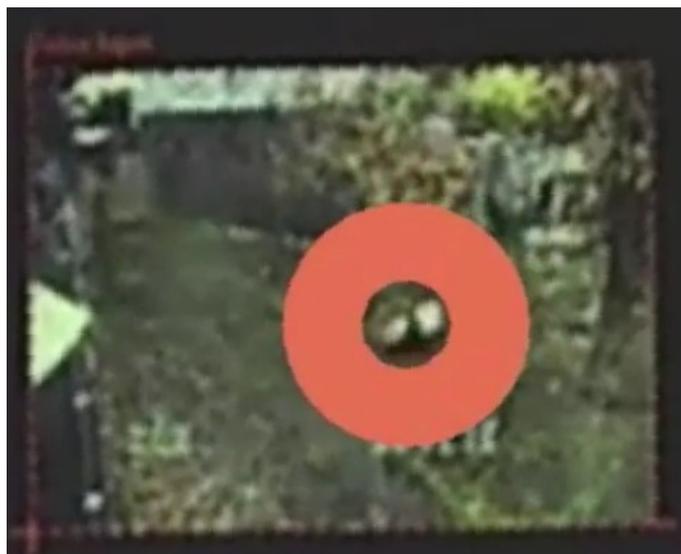


Figura 64: Resultado obtido com o eye-tracker.



Figura 65: Resultado obtido com o Sistema de Rastreamento Visual.

Foi utilizado, para realização deste experimento, 150 imagens do vídeo Panda disponível no *benchmark* do TLD. Uma representação estática em 2D do resultado obtido com o *eye-tracker* pode ser vista na Figura 66 e um resumo do resultado obtido com o Sistema de Rastreamento Visual proposto pode ser vista na Figura 67 (o resultado do sistema foi resumido dado que temos como saída do sistema um resultado por *frame*, enquanto os resultados humanos são em quantidade bem inferiores). Podemos perceber que os caminhos resultantes são muito similares.

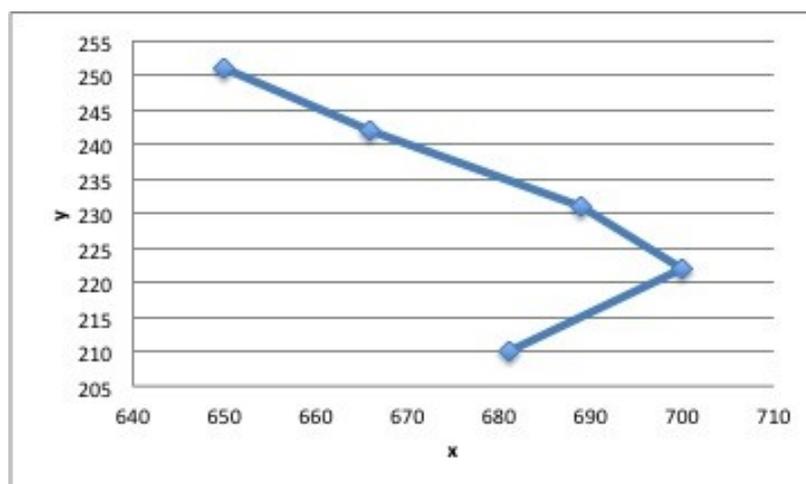


Figura 66: Gráfico com os resultados obtidos com o eye-tracker.

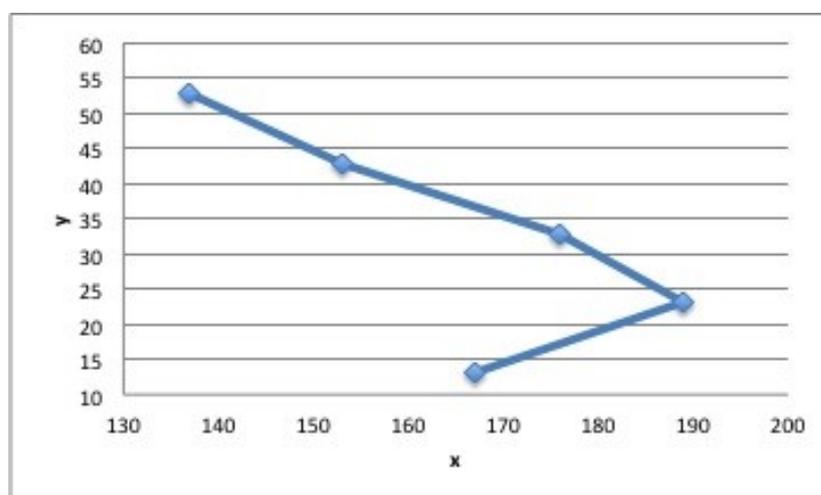


Figura 67: Gráfico com os resultados obtidos com o sistema de rastreamento visual proposto.

A partir deste experimento preliminar, realizaremos novos experimentos buscando aprofundar nossos testes com o *eye-tracker* a fim de comparar/validar/aperfeiçoar o Sistema de Rastreamento Visual. Buscaremos, ainda, modelar o movimento ocular de perseguição suave, intimamente interligado ao movimento de sacada ocular.

6 CONCLUSÃO

Neste trabalho, apresentamos uma modelagem matemático-computacional de uma arquitetura neural que representa o Superior Colliculus. Esta arquitetura neural é baseada em Generalização Virtual de Memória de Acesso Aleatório em Redes Neurais Sem Peso (*Virtual Generalizing Random Access Memory Weightless Neural Networks – VGRAM WNN*) e no mapeamento log-polar da retina para o Superior Colliculus. Com a nossa implementação desta arquitetura é possível, a partir de pontos de interesse em uma determinada imagem bidimensional previamente treinados, realizar a busca visual por estes pontos em imagens diferentes da treinada. O modelo de busca visual biologicamente inspirado foi incorporado em um sistema automático de rastreamento (*tracking*) de longo prazo de objetos de interesse em vídeo e os resultados obtidos com este rastreador se equiparam ao estado da arte.

Para realizarmos a busca visual de um objeto de interesse em uma imagem utilizamos um modelo de neurônios de RNSP do tipo VG-RAM com memória compartilhada. Estes neurônios se conectam ao plano da imagem por meio de uma distribuição sináptica log-polar (da mesma forma que é feita no sistema visual humano), que enfatiza a região ao redor do ponto de atenção e atenua a representação da região periférica. O modelo de memória compartilhada que empregamos se justifica pela capacidade dos neurônios do Superior Colliculus fornecerem um padrão de ativação independentemente da posição do objeto no plano da imagem, segundo sua projeção em V1.

Incorporamos o modelo de busca visual biologicamente inspirado em um sistema de rastreamento de longo prazo. Realizamos uma série de experimentos de calibração dos parâmetros do modelo visando obter a configuração que fornecesse o melhor desempenho em termos de taxa de acerto com o menor custo computacional possível.

Com os melhores conjuntos de parâmetros obtidos, avaliamos o desempenho do sistema de rastreamento de objetos utilizando o banco de dados TLD e um vídeo com um resumo dos resultados alcançados está disponível em <https://www.youtube.com/watch?v=rz5-5IG6yU>. Realizamos, também, um experimento do tipo “siga-o-líder” com o carro autônomo IARA do Laboratório de Computação de Alto Desempenho (LCAD) da Universidade Federal do Espírito Santo (UFES) e um vídeo com o experimento está disponível em <https://www.youtube.com/watch?v=lePu4KskvNk>. Realizamos, por fim, um experimento preliminar com um *eye-tracker* e um vídeo com este experimento está disponível em <https://www.youtube.com/watch?v=MIMlxHp4uz0>. Os resultados experimentais mostraram que a abordagem proposta é capaz de rastrear de maneira confiável e eficiente uma grande variedade de objetos.

6.1 Trabalhos Futuros

Como trabalho futuro, gostaríamos de aproximar ainda mais o nosso sistema de rastreamento visual do sistema visual biológico dos mamíferos, modelando a forma com que detectamos um objeto quando este sai de cena e o movimento ocular de perseguição suave. Uma ferramenta importante nestes trabalhos será o *eye-tracker*.

Também está prevista a implementação de uma versão paralela do modelo sugerido para a execução em GPGPUs - *General Purpose Graphics Processing Units* - em linguagem CUDA - *Compute Unified Device Architecture* - ou mesmo em outros dispositivos aceleradores, possibilitando elevado ganho de desempenho e viabilizando a sua execução em aplicações de tempo real.

Outras importantes melhorias no sistema de rastreamento visual atual são: não manter o mesmo aspecto inicial da caixa delimitadora do objeto de interesse ao longo de todo o vídeo, melhorar a técnica de detecção do objeto após oclusão e fazer com que os limiares de retreino sejam dependentes das características de cada vídeo.

Por fim, como trabalho futuro gostaríamos de estender este trabalho para rastreamento de múltiplos objetos de interesse.

REFERÊNCIAS BIBLIOGRÁFICAS

1. YILMAZ, A.; JAVED, O.; SHAH, M. Object Tracking: A Survey. **ACM Computing Surveys**, 2006.
2. FUA, P.; LEPETIT, V. Monocular model-based 3D tracking of rigid objects. **Comput. Graph. Vis.**, v. 1, n. 1, p. 1-89, 2005.
3. GOMES, H. M.; ZHANG, T. Technology survey on video face tracking. **Imaging and Multimedia Analytics in a Web and Mobile World 2014**, 2014.
4. LUCAS, B. D.; KANADE, T. **An iterative image registration technique with an application to stereo vision**. Proceedings of the International Joint Conference on Artificial Intelligence, pages 674–679. [S.l.]: [s.n.]. 1981.
5. COMANICIU, D.; RAMESH, V.; MEER, P. **Real-time tracking of non-rigid objects using mean shift**. In IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pages 142–149. [S.l.]: [s.n.]. 2000.
6. OZUYSAL, M.; FUA, P.; LEPETIT, V. **Fast keypoint recognition in ten lines of code**. IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos, CA, USA: [s.n.]. 2007.
7. AVIDAN, S. Support vector tracking. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 26, n. 8, p. 1064-1072, 2004.
8. COLLINS, R. T.; LIU, Y.; LEORDEANU, M. Online selection of discriminative tracking features. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 27, n. 10, p. 1631-1643, 2005.
9. JAVED, O.; ALI, S.; SHAH, M. **Online detection and classification of moving objects using progressively improving detectors**. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), volume 1, pages 696–701. [S.l.]: [s.n.]. 2005. p. 696-701.
10. ROSS, D. et al. Incremental learning for robust visual tracking. **International Journal of Computer Vision**, v. 77, n. 1, p. 125-141, 2008.
11. ADAM, A.; RIVLIN, E.; SHIMSHONI, I. **Robust fragments-based tracking using the integral histogram**. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR06), volume 1, pages 798–805. [S.l.]: [s.n.]. 2006.

12. AVIDAN, S. Ensemble tracking. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 29, n. 2, p. 261-271, 2007.
13. GRABNER, H.; LEISTNER, C.; BISCHOF, H. **Semi-supervised On-Line boosting for robust tracking**. Proceedings of the 10th European Conference on Computer Vision, volume 5302, pages 234–247. Berlin, Heidelb: [s.n.]. 2010.
14. BABENKO, B.; YANG, M.-H.; BELONGIE, S. **Visual tracking with online multiple instance learning**. IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops), pages 983–990. [S.I.]: [s.n.]. 2009.
15. STALDER, S.; GRABNER, H.; VAN GOOL, L. **Beyond semi-supervised tracking: Tracking should be as simple as detection, but not simpler than recognition**. IEEE International Conference on Computer Vision Workshops, pages 1409–1416. [S.I.]: [s.n.]. 2009.
16. SANTNER, J. et al. **PROST: Parallel robust online simple tracking**. IEEE Conference on Computer Vision and Pattern Recognition, pages 723–730. [S.I.]: [s.n.]. 2010.
17. HARE, S.; SAFFAR, A.; TORR, P. H. S. **Struck: Structured output tracking with kernels**. IEEE International Conference on Computer Vision, pages 263–270. [S.I.]: [s.n.]. 2011.
18. KALAL, Z.; MIKOLAJCZYK, K.; MATAS, J. Tracking-Learning-Detection. **IEEE Transactions On Pattern Analysis And Machine Intelligence**, v. 6, n. 1, 2010.
19. GUO, P. et al. **Adaptive and accelerated tracking-learning-detection**. Proc. of International Symposium on Photoelectronic Detection and Imaging. [S.I.]: [s.n.]. 2013.
20. CAO, W. et al. **A multiple face detection and tracking system based on TLD**. Proceedings of the Fifth International Conference on Internet Multimedia Computing and Service, pp. 386-389. [S.I.]: [s.n.]. 2013.
21. PAN, X. Q.; ZHANG, H. F. **Research on Target Tracking Based on TLD Algorithm**. Applied Mechanics and Materials, pp. 819-822. [S.I.]: [s.n.]. 2013.
22. ALEKSANDER, I. RAM-Based Neural Networks From WISARD to MAGNUS: a Family of Weightless Virtual Neural Machines. In: AUSTIN, J. **RAM-Based Neural Networks**. [S.I.]: World Scientific, 1998. p. 18-30.
23. LUDERMIR, T. B. et al. Weightless neural models: a review of current and past works. **Neural Computing Surveys**, v. 2, 1999. ISSN 41-61.
24. DE SOUZA, A. F. et al. **Face Recognition with VG-RAM Weightless Neural**

- Networks**. Artificial Neural Networks - ICANN 2008. Berlin: [s.n.]. 2008.
25. MORAES, J. L.; DE SOUZA, A. F.; BADUE, C. **Facial access control based on VG-RAM weightless neural networks**. International Conference on Artificial Intelligence (ICAI 2011). [S.l.]: [s.n.]. 2011.
 26. DE SOUZA, A. F. et al. Automated multi-label text categorization with VG-RAM weightless neural networks. **Neurocomputing**, v. 72, n. 2, p. 2209-2217, 2009.
 27. DE SOUZA, A. F.; FREITAS, F. D.; COELHO, A. G. C. D. Fast learning and predicting of stock returns with virtual generalized random access memory weightless neural networks. **Concurrency and Computation**, v. 24, n. 8, p. 921-933, 2012.
 28. BERGER, M. et al. **Traffic sign recognition with VG-RAM Weightless Neural Networks**. 12th International Conference on Intelligent Systems Design and Applications (ISDA). [S.l.]: [s.n.]. 2012.
 29. BERGER, M. et al. Traffic Sign Recognition with WISARD and VG-RAM Weightless Neural Networks. **Journal of Network and Innovative Computing**, v. 1, p. 87-98, 2013.
 30. DE SOUZA, A. F. et al. **Traffic sign detection with VG-RAM weightless neural networks**. International Joint Conference on Neural Networks. [S.l.]: [s.n.]. 2013.
 31. STAFFA, M. et al. **Can you follow that guy?** 22th European Symposium on Artificial Neural Networks - ESANN. [S.l.]: [s.n.]. 2014. p. 511-516.
 32. MOREIRA, R. S. et al. **Tracking targets in sea surface with the WiSARD weightless neural network**. BRICS Conference on Computational Intelligence, IEEE. [S.l.]: [s.n.]. 2013. p. 166-171.
 33. SILVA MOREIRA, R. D.; FAVILLA EBECKEN, N. F. Parallel wisard object tracker: A ram-based tracking system. **Computer Science & Engineering: An International Journal**, v. 4, p. 1-13, 2014.
 34. KANDEL, E. R. . S. J. H. . J. T. M. **Principles of Neural Science**. 4th Ed. ed. [S.l.]: Prentice-Hall International, Inc., 2000.
 35. SCHOLARPIDIA. Retina, dez. 2014. Disponivel em: <<http://www.scholarpedia.org/article/Retina>>. Acesso em: 2014.
 36. SUSANA MARTINEZ-CONDE, J. O.-M. S. L. M. The impact of microsaccades on vision: towards a unified theory of saccadic function. **Nature Reviews Neuroscience**, p. 83-96, 2013.

37. OYSTER, C. W. **The Human Eye: Structure and Function**. [S.l.]: Sinauer Associates , 1999.
38. TOOTELL, R. B. et al. Deoxyglucose analysis of retinotopic organization in primate striate cortex. **Science**, n. 218, p. 902-904, 1982.
39. MUNOZ, D. P.; D., P.; D, G. Movement of Neural Activity on the Superior Colliculus. **Science**, v. 251, p. 1358-1360, 1991. ISSN 4999.
40. KLIER, E. M.; WANG, H.; CRAWFORD, J. D. The Superior Colliculus Encodes Gaze Commands in Retinal Coordinates Nature Neuroscience. **Nature Neuroscience**, v. 4, p. 627-632, 2001.
41. EBENHOLTZ, S. M. **Oculomotor Systems and Perception**. [S.l.]: Cambridge University Press, 2001.
42. NETO, J. O. **Um Sistema de Busca Visual Biologicamente Plausível Baseado em Redes Neurais Sem Peso**. Universidade Federal do Espírito Santo. Vitória. 2012.
43. SPARKS, D. L.; GANDHI, N. J. Single cell signals: an oculomotor perspective. **Progress in Brain Research**, v. 142, 2003.
44. SOETEDJO, R.; KANEKO, C. R. S.; FUCHS, A. F. Evidence Against a Moving Hill in the Superior Colliculus During Saccadic Eye Movements in the Monkey. **Journal of Neurophysiology**, v. 87, p. 2778-2789, 2002.
45. FONTANA, C. M. C. **Movimentos Sacádicos Virtuais Baseados em VG-RAM WNN**. PPGI - UFES. Vitória. 2013.
46. MARINO, R. A. et al. Spatial Interactions in the Superior Colliculus predict saccade behavior in a neural field model. **Journal of Cognitive Neuroscience**, v. 24, n. 2, p. 315-336, 2012.
47. KANDELL, E. R. et al. **Principles of Neural Science**. [S.l.]: Prentice-Hall International, 2000.
48. GREGORY, R. **Eye and Brain**. 5th Ed. ed. [S.l.]: Oxford University Press, 1998.
49. FARDIN JR, D.; KOMATI, K. S.; DE SOUZA, A. F. **Arquitetura do Sistema Visual Humano: Uma Abordagem Computacional**. III Encontro Regional de Informática RJ/ES. Porto Alegre - RS: Sociedade Brasileira de Computação. 2003. p. 95-126.

50. HUBEL, D. H. **Eye, Brain and Vision**. [S.l.]: Scientific American Library, 1995.
51. ALEKSANDER, I. Self-adaptive universal logic circuits. **IEEE Electronic Letters**, v. 2, n. 8, p. 231-232, 1966.
52. ERCEGOVAC, M. D.; LANG, T.; MORENO, J. H. **Introduction to Digital Systems**. [S.l.]: Wiley, 1998.
53. FORECHI, A. **Sistema de Navegação Robótica por Imagens de Pontos de Interesse**. PPGI - UFES. Vitória. 2012.
54. KOMATI, K. S.; DE SOUZA, A. F. Vergence Control in a Binocular Vision System using Weightless Neural Networks. **Proceedings of the 4th International Symposium on Robotics and Automation**, 2002.
55. DE SOUZA, A. F. et al. Face Recognition with VG-RAM Weightless Neural Networks. **Lecture Notes in Computer Science**, v. 5163, n. 1, p. 951-960, 2008.
56. DE SOUZA, A. F. et al. Improving VG-RAM WNN Multi-label Text Categorization via Label Correlation. **8th International Conference on Intelligent Systems Design and Applications**, 2008.
57. DE SOUZA, A. F. et al. Automated Multi-label Text Categorization with VG-RAM Weightless Neural Networks. **Neurocomputing**, v. 72, p. 2209-2217, 2009.
58. DE SOUZA, A. F. et al. Automated Free Text Classification of Economic Activities using VG-RAM Weightless Neural Networks. **Proceedings of the 7th International Conference on Intelligent Systems Design and Applications**, 2007. 782-787.
59. BADUE, C.; PEDRONI, F.; DE SOUZA, A. F. Multi-Label Text Categorization using VG-RAM Weightless Neural Networks. **Proceedings of the 10th Brazilian Symposium on Neural Networks (SBRN'08)**, Salvador, BA, 2008. 105-110.
60. DE SOUZA, A. F. et al. High Performance VG-RAM WNN. **Submetido para publicação**, 2014.
61. BERGER, M. et al. Visual Tracking with VG-RAM Weightless Neural Networks. **Neurocomputing**, Amsterdam, 2015. ISSN 0925-2312.
62. MITCHELL, R. J. et al. **Comparison of Some Methods for Processing Grey Level Data in Weightless Networks**. [S.l.]: World Scientific, 1998. 61-70 p.

63. **Máquina Associadora de Eventos**, 2010. Disponível em: <http://www.lcad.inf.ufes.br/wiki/index.php/Máquina_Associadora_de_Eventos_-_MAE>.
64. **Laboratório de Computação de Alto Desempenho**. Disponível em: <<http://www.lcad.inf.ufes.br>>.
65. OLIVEIRA, H. **Uma Modelagem Computacional de Áreas Corticais do Sistema Visual Humano Associadas à Percepção de Profundidade**. PPGI - Universidade Federal do Espírito Santo. Vitória. 2005.
66. WINTER, D.; LEVANDOWSKY, M. Distance between sets. **Nature**, v. 234, n. 5323, p. 34-35, 1971.
67. KALAL, Z.; MIKOLAJCZYK, K.; MATAS, J. Tracking-Learning-Detection. **IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE**, v. 6, n. 1, Janeiro 2010.
68. **TORC Robotics**. Disponível em: <<http://www.torcrobotics.com/>>. Acesso em: 01 jun. 2014.
69. MONTEMERLO, M.; ROY, N.; THRUN, S. Perspectives on standardization in mobile robot programming: The carnegie mellon navigation (CARMEN) toolkit. **In Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)**, 2003. 2436-2441.
70. MONTEMERLO, M.; ROY, N.; THRUN, S. **Perspectives on standardization in mobile robot programming**: The carnegie mellon navigation (CARMEN) toolkit. International Conference on Intelligent Robots and Systems (IROS). [S.l.]: [s.n.]. 2003. p. 2436-2441.
71. SIMMONS, R. The inter-process communication (IPC) system. Disponível em: <<http://www.cs.cmu.edu/afs/cs/project/TCA/www/ipc/ipc.html>>.
72. WALTON, M. M. G.; SPARKS, D. L.; GANDHI, N. J. Simulations of Saccade Curvature by Models That Place Superior Colliculus Upstream From the Local Feedback Loop. **Journal of Neurophysiology**, 2005. 2354-2358.
73. PORT, N. L.; WURTZ, R. H. Sequential activity of simultaneously recorded neurons in the superior colliculus during curved saccades. **Journal of Neurophysiology**, 2003. 1887-1903.