PONTOS DE INTERESSE ESPAÇO-TEMPORAL PARA RECONHECIMENTO DA LÍNGUA BRASILEIRA DE SINAIS EM VÍDEOS

JOSIANE DE OLIVEIRA FAGUNDES DE ARAÚJO

PONTOS DE INTERESSE ESPAÇO-TEMPORAL PARA RECONHECIMENTO DA LÍNGUA BRASILEIRA DE SINAIS EM VÍDEOS

Dissertação apresentada ao Programa de Pós-Graduação em Modelagem Computacional e Sistemas da Universidade Estadual de Montes Claros como requisito parcial para a obtenção do título de Mestre em Modelagem Computacional e Sistemas.

Orientador: Prof. Dr. Antônio Wilson Vieira

Montes Claros

Agosto de 2017

de Oliveira Fagundes de Araújo, Josiane

D1234p Pontos de interesse espaço-temporal para reconhecimento da Língua Brasileira de Sinais em vídeos / Josiane de Oliveira Fagundes de Araújo. — Montes Claros, 2017
xx, 51 f. : il. ; 29cm

Dissertação (mestrado) — Universidade Estadual de Montes Claros
Orientador: Prof. Dr. Antônio Wilson Vieira

1. Computação — Teses. 2. Visão Computacional — Teses. I. Orientador. II. Título.

CDU 519.6*82.10

Unimontes

RELATÓRIO DE DEFESA DE DISSERTAÇÃO

MESTRADO

Universidade Estadual de Montes Claros - Unimontes
Pró-Reitoria de Ensino/Coordenadoria de Pós-Graduação
Programa de Pós-Graduação em Modelagem Computacional e Sistemas - PPGMCS

1 - Identificação do Aluno

Nome: Josiane de Oliveira Fagundes da Silva	Matrícula: 9992833
Linha de Pesquisa: Inteligência Computacional, Otimização e suas Aplicações.	

2 - Sessão de Qualificação

Título: **"PONTOS DE INTERESSE ESPAÇO-TEMPORAL PARA RECONHECIMENTO DA LÍNGUA BRASILEIRA DE SINAIS EM VÍDEOS"**

3 - Comissão Examinadora

Nome	Função	Assinatura
Prof. Dr. Antônio Wilson Vieira	Orientador (a)	
Prof. Dr. Marcos Flávio S. V. D'ângelo	Examinador(a)	
Prof. Dr. Renê Rodrigues Velosos	Examinador(a)	

4 - Resultado

A comissão Examinadora, em **02/08/2017** após Defesa de Dissertação e arguição do(a) candidato(a), decidiu:

( ) pela aprovação da Dissertação
( ) pela reprovação da Dissertação
(X) pela revisão de forma, indicando o prazo de 30 dias para apresentação definitiva.
( ) pela reformulação da Dissertação, indicando o prazo de _____ dias para nova versão.

Preencher somente em caso de revisão de forma:	**Preencher somente em caso de revisão de reformulação:**
(X) O(a) aluno(a) apresentou a revisão de forma e a Dissertação foi aprovada.	( ) O(a) aluno(a) apresentou a reformulação e a Dissertação foi aprovada.
( ) O(a) aluno(a) apresentou a revisão de forma e a Dissertação foi reprovada.	( ) O(a) aluno(a) apresentou a reformulação e a Dissertação foi reprovada.
( ) O(a) aluno(a) não apresentou a revisão da forma.	( ) O(a) aluno(a) não apresentou a reformulação.

Autenticação
Orientador(a) Comissão Examinadora

02/08/2017
Data

Assinatura

Autenticação
Coordenador

02/08/2017
Data

Prof. Nilton Alves Maia
Coordenador do PPGMCS
UNIMONTES
Matrícula 1046657-1

Assinatura

Dedico este trabalho aos meus filhos, Victor e Heitor, razão do meu viver e minha maior alegria.

Agradecimentos

A Deus, pelos cuidados nessa jornada tão difícil.

Aos meus pais, Laurinda e Argemiro, e irmãos, Juliana e Guilherme, pela paciência, compreensão e apoio incondicional nos momentos que eu mais precisei.

Aos meus filhos, Victor e Heitor, por suportarem, nem sempre com paciência, minha ausência nos momentos em que a dedicação ao mestrado era necessária.

Aos meus amigos surdos, Claudiane, Jacson Rafael, Maria Franciel, Rubens e Camila, pela colaboração neste trabalho e pela amizade sincera de tantos anos.

À minha companheira do lar, Gislene, pelos cuidados dispensados a mim e aos meus filhos durante esse percurso.

Ao meu marido, Jackson, por aparecer em minha vida no meio dessa jornada e torná-la muito mais leve e agradável.

Aos professores Marcos Flávio e Renê, pela parceria, ensinamentos e participação na avaliação deste trabalho.

Especialmente ao meu orientador, Professor Antônio Wilson, pelos conhecimentos compartilhados e pelas palavras duras, motivadoras, de apoio, de consolo, sempre nos momentos em que eu mais precisava.

Meu muito obrigada por acreditarem em mim e vivenciarem comigo o sonho de realização desse trabalho.

Resumo

Este trabalho aborda o reconhecimento de sinais da Libras usando apenas sequências de vídeo, que são acessíveis e disponíveis na maioria dos dispositivos móveis, como *smartphones* e *tablets*. A abordagem utilizada extrai pontos de interesse espaço-temporal em sequências de vídeo para construir um dicionário visual de forma que os vídeos são descritos em termos desse dicionário. Um conjunto de 1500 vídeos, com até 100 sinais diferentes, realizados por vários sujeitos, que são surdos nativos ou fluentes nessa língua de sinais, foi produzido para validar o método proposto. O trabalho explora ajuste de parâmetros para construção de descritores nessa base de vídeos e considera o particionamento da base para explorar a classificação com grupos de sinais com maior ou menor similaridade. Os resultados experimentais mostram que o método produz descritores capazes de obter altas taxa de classificação considerando classificadores usuais da literatura, especialmente com o LDA. Além disso, o trabalho considera o método de classificação Imune/neural, baseado na associação de sistemas imunes com uma rede neural que, apesar de exigir maior tempo de processamento, apresenta resultados superiores ao LDA.

Palavras-chave: Reconhecimento de ações humanas, Libras, pontos de interesse espaço-temporal.

Abstract

This work addresses the recognition of Libras signals using only video sequences, which are accessible and available on most mobile devices such as smart phones and tablets. The approach used extracts spatio temporal interest points from the video sequences to construct a visual dictionary so that videos are described in terms of that dictionary. A set of 1500 videos, with up to 100 different signals, performed by several subjects, who are native deaf or have fluence in that signal language, was produced to validate the proposed method. The work explores the adjustment of parameters for constructing the descriptors in this video database and considers the base partitioning to explore classification with groups of signals with greater or lesser similarity. The experimental results show that the method produces descriptors capable of obtaining high classification rate considering usual classifiers from the literature, especially with LDA. In addition, the work considers the Immune/neural method for classification based on the association of immune systems with a neural network that, despite requiring more processing time, presents superior results in comparison with LDA.

Keywords: Human action recognition, Libras, space-time key points.

Lista de Figuras

Lista de Tabelas

Sumário

Capítulo 1

Introdução

No Brasil, milhões de pessoas são surdas ou possuem deficiência auditiva severa. Devido à dificuldade de se comunicar em Língua Portuguesa (LP), que depende da capacidade de ouvir, muitos deles utilizam uma língua gestual visual chamada Língua Brasileira de Sinais (Libras). A Lei de Libras [Brasil, 2002] e o Decreto 5.626 [Brasil, 2005] estabelecem que a Libras é a primeira língua para comunicação dos surdos no país. Além disso, definem diretrizes para o ensino da Libras no contexto escolar e utilização de intérpretes em vários ambientes públicos, para intermediar a comunicação entre surdos e pessoas que possuem plena capacidade de ouvir e se comunicar com uma língua oral, denominados ouvintes.

Assim como a língua oral, diversas nações desenvolveram sua própria língua de sinais. A Libras, como toda língua de sinais, é uma língua de modalidade gestual-visual que utiliza como canal de comunicação gestos e expressões que são percebidos pela visão. Já a Língua Portuguesa, que é uma língua de modalidade oral-auditiva, utiliza como canal sons articulados que são percebidos pelos ouvidos [Souza et al., 2016]. Entretanto, existem semelhanças entre elas, como a existência de unidades mínimas formadoras de unidades complexas. Isso pode ser observado em todas as línguas de sinais, que são diferentes em cada país. Ambas possuem os níveis fonológico, morfológico, sintático, semântico e pragmático, assim como as línguas orais. No Brasil, onde Libras é a primeira língua para comunicação dos surdos, a Língua Portuguesa deve ser aprendida por eles como segunda língua e na forma escrita [Brasil, 2005].

Nos diversos ambientes frequentados por surdos e ouvintes que não são fluentes em Libras, faz-se necessário a presença de um tradutor/intérprete de Libras/LP para intermediar a comunicação. Assim, o intérprete é responsável por traduzir o que está sendo dito em LP para Libras e vice versa. No entanto, nem sempre é verificado a disponibilidade desse profissional em todos os locais frequentados por surdos e muitas

vezes o conteúdo da conversa entre um surdo e um ouvinte pode ser confidencial, o que torna inapropriada a participação de um terceiro.

Com a popularização dos meios de comunicação, torna-se fácil acessar a *web* a partir de dispositivos móveis, viabilizando a utilização desse meio para comunicação com pessoas surdas, sendo que ambos os interlocutores devem dominar a LP escrita ou a Libras. No caso da comunicação por texto, existem vários aplicativos de mensagem disponíveis. No entanto, há grande dificuldade de compreensão da LP escrita por parte dos surdos, já que ainda é um desafio educacional o ensino de português como segunda língua para eles. Por outro lado, poucos ouvintes dominam a Libras a ponto de se comunicarem fluentemente com surdos, apesar da recente propagação da língua principalmente no meio educacional.

No caso em que um dos interlocutores conhece apenas a Libras e o outro apenas a LP, surgem dois desafios tecnológicos para permitir a comunicação sem a presença de intérpretes. O primeiro trata-se de traduzir a LP para Libras e o segundo do problema inverso, isto é, de traduzir da Libras para a LP.

A tradução da LP escrita para Libras é realizada na produção de animações gráficas, a partir de texto ou áudio em LP, com gestos produzidos por um *avatar* na tela do computador ou dispositivo móvel de forma que o surdo compreenda em Libras a mensagem originalmente fornecida em LP. Para essa direção da comunicação, existem algumas soluções como o *WebLibras* e o *HandTalk* [Wanderlan et al., 2013]. A Figura 1.1 ilustra o Hugo, *avatar* do *HandTalk*, sinalizando a palavra *oi*.

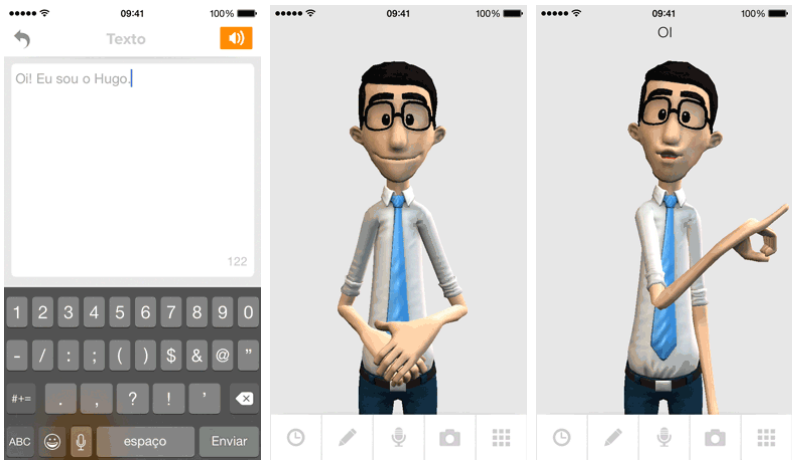
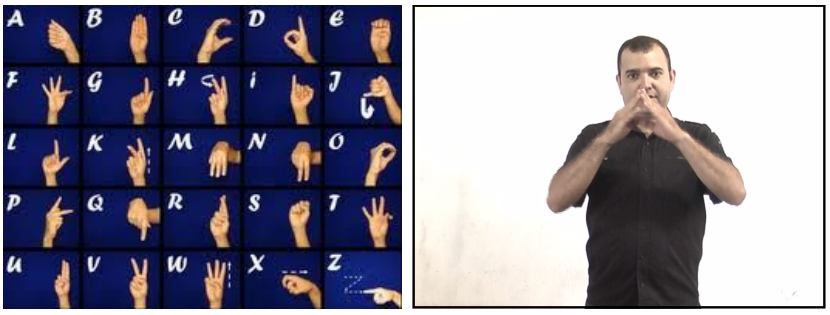


Figura 1.1. Imagens criadas através do aplicativo *HandTalk*.

Na direção oposta, a tradução da Libras para a LP, verificamos que é um problema mais desafiador devido à necessidade de capturar em vídeo, ou outro dispositivo, a mensagem sinalizada e traduzir para LP escrita. Nesse sentido, existem muitas inici-

ativas e pesquisas, mas ainda não está disponível uma solução completa. Além disso, a maioria das pesquisas tem se concentrado em sinais estáticos e no reconhecimento de letras e números, o que é um escopo muito pequeno da Libras. Nesse caso, uma imagem é suficiente para o reconhecimento do sinal ou letra. No entanto, a Libras possui ainda um vasto vocabulário composto por sinais dinâmicos, isto é, que possuem uma combinação de configuração de mãos, movimentos e expressões faciais e corporais.

As Figuras 1.2 e 1.3 ilustram a diferença entre simples alfabeto manual e sinais estáticos ou dinâmicos. Enquanto que o alfabeto manual é utilizado tão somente na soletração de nomes próprios ou quando se deseja explicitar a palavra em LP, os sinais estáticos ou dinâmicos indicam palavras ou expressões completas da Libras. Os sinais estáticos, como o sinal de *casa* apresentado na Figura 1.2, podem ser reconhecidos utilizando uma única imagem, pois não possuem movimento incorporado. Já os sinais dinâmicos, ilustrados na Figura 1.3, possuem movimentos dos braços e mãos, além de expressões faciais e corporais. Nesse caso, a sequência de imagens representa alguns quadros do movimento, já que o vídeo é composto por 30 quadros por segundo.



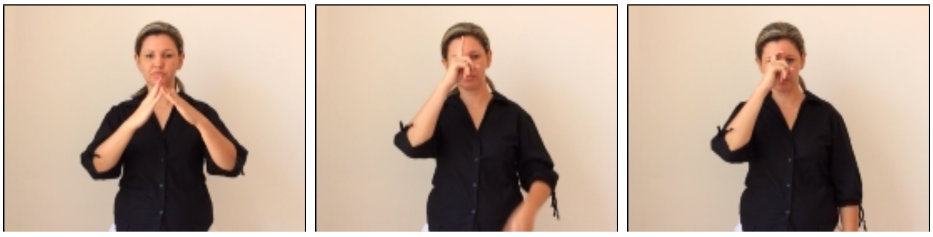
a) Alfabeto manual b) Sinal de *casa*

Figura 1.2. Exemplo de alfabeto manual e sinais estáticos. Em (a) ilustramos o alfabeto manual de Libras (fonte: http://portaldoprofessor.mec.gov.br) e, em (b) um sinal estático articulado para a palavra *casa*, em Libras.

Uma abordagem de reconhecimento de sinais usando vídeos é desafiadora e interessante, pois deve ser capaz de capturar a dinâmica dos movimentos e configurações de mãos. No entanto, uma abordagem regularmente encontrada nos trabalhados pesquisados são soluções que consideram o uso de luvas ou outros sensores, como o sensor de profundidade. Isso torna a solução menos aplicável ao público em geral, devido à possível dificuldade de acesso a tais equipamentos.

Nesse contexto, este trabalho visa contribuir com o desenvolvimento de aplicativos que permitam a tradução automática de sinais da Libras em vídeo para texto em LP, sem a utilização de luvas ou outros sensores. Para isso, uma abordagem que utiliza técnicas de Visão Computacional e reconhecimento de padrões foi utilizada. Essa

a) Sinal de *igreja*



b) Sinal de *hospital*

Figura 1.3. Exemplos de sinais dinâmicos. Em (a) três quadros do sinal de *igreja* e, em (b), três quadros do sinal de *hospital*.

abordagem consiste na detecção de pontos de interesse espaço-temporal em vídeos, objetivando a captura da dinâmica espacial e temporal dos sinais. Em seguida, é aplicada uma técnica de agrupamento para detecção de pontos de interesse similares, formando um vocabulário visual a partir dos grupos detectados. De posse desse vocabulário, cada vídeo é representado através da contagem de palavras visuais, definidas pelos grupos, gerando vetores descritores dos vídeos. Tais descritores são usados para treinamento do classificador e testes. Nessa etapa, alguns classificadores disponíveis no *Scikit-Learn* são testados e a Análise Discriminante Linear (LDA), que obtém o melhor resultado, é comparado com o classificador Imune/neural [D'Angelo et al., 2016], que segue uma abordagem evolutiva.

1.1 Objetivos

De forma geral, o objetivo desse trabalho é contribuir na direção do reconhecimento de sinais da Libras em vídeos para viabilizar o desenvolvimento de sistemas tradutores de Libras para Língua Portuguesa, de forma a dar suporte ao desenvolvimento de sistemas que favoreçam a inclusão social de pessoas surdas. Assim, são estabelecidos os seguintes objetivos específicos:

- Criar uma base de vídeos de sinais da Libras, selecionados a partir de critérios linguísticos, a ser usada para validar a abordagem deste trabalho, bem como

servir de referência para trabalhos futuros.

- Utilizar técnicas de reconhecimento de padrões em vídeo para detectar e descrever de forma vetorial os pontos de interesse que caracterizam a dinâmica espacial e temporal de cada tipo de sinal.

- Descrever cada vídeo em termos de pontos de interesse detectados de forma a viabilizar a classificação dos sinais em vídeos.

1.2 Contribuições

A principal contribuição deste trabalho foi propor a utilização de uma abordagem para construir um modelo de classificação de sinais da Libras que usa apenas sequências de vídeo como entrada, sem a necessidade de luvas ou sensores especiais. Dessa forma, a solução proposta é de fácil acesso, pois pode ser aplicável a uma câmera embutida de dispositivos móveis, por exemplo. Assim, a abordagem utilizada contribui para o desenvolvimento de aplicativos para dispositivos móveis, que torne possível a tradução de Libras para LP escrita em qualquer ambiente e a qualquer momento.

Outra contribuição importante do trabalho foi que a abordagem utilizada não considerou apenas sinais estáticos, com uma única pose, ou letras e números. Além disso, reconheceu sinais dinâmicos, compostos de movimentos articulados das mãos e expressões não manuais. Para isso, usou pontos de interesse espaço-temporal para capturar a dinâmica espacial e temporal dos sinais. Esse foi um ponto importante pois, além de ter sido pouco explorado na literatura, a maior parte da comunicação através da Libras utiliza sinais dinâmicos e não apenas letras do alfabeto manual.

Na etapa de classificação, uma formulação híbrida Imune/neural, apresentada em [D'Angelo et al., 2016], foi usada para aumentar o desempenho de classificação em comparação com o método de Análise de Discriminante Linear. Esse método de classificação é composto pela associação de ClonALG, baseado em sistemas imunológicos [de Castro & Zuben, 2002], com a rede neural de Kohonen [Kohonen, 2001]. Além disso, para combinar estas duas abordagens, os critérios de parada e o mecanismo de seleção tiveram que ser alterados para acomodar um número não fixo de anticorpos. Para esta abordagem, não são necessários modelos matemáticos ou estatísticos, diminuindo o problema de complexidade de implementação da solução.

O desenvolvimento da pesquisa contribuiu ainda com a produção de um conjunto de dados de vídeo com 100 sinais diferentes da Libras e um total de 1500 amostras de vídeos. Esses sinais foram escolhidos por especialistas em Libras, propositalmente

com pequenas e grandes variações linguísticas, possibilitando a verificação de grupos de sinais com escala crescente de dificuldade no processo de classificação. Além disso, os sinais foram realizados por sujeitos fluentes em Libras e surdos nativos. Sendo assim, espera-se que os vídeos sejam amostras fiéis da Libras, incorporando a naturalidade da sinalização com suas variações linguísticas esperadas.

1.3 Publicações

Durante o desenvolvimento deste trabalho, foram produzidos dois artigos científicos, sendo um deles publicado em congresso nacional e outro submetido para uma revista internacional. Além disso, um terceiro artigo em congresso nacional foi produzido com nossa colaboração:

- Josiane O. Fagundes, Gustavo W. Ferreira, Antonio W. Vieira. Pontos de interesse espaço-temporal para reconhecimento da Libras, XIX ENMC Encontro Nacional de Modelagem Computacional, João Pessoa-PB, Out/2016.
- Josiane O. Fagundes, Antonio W. Vieira, Marcos F.S. D'Angelo. Brazilian Sign Language recognition from space-time interest points and Immune/neural classification. Submetido para Applied Soft Computing.
- Gustavo W. Ferreira, Josiane O. Fagundes, Antonio W. Vieira. Reconhecimento de ações humanas em dados do Kinect usando matriz de distância e dicionário visual. XIX ENMC Encontro Nacional de Modelagem Computacional, João Pessoa-PB, Out/2016.

1.4 Organização da dissertação

Este trabalho está assim organizado: O Capítulo 2 apresenta uma revisão de alguns trabalhos relacionados ao reconhecimento de ações em geral e, especialmente, sobre reconhecimento das línguas de sinais. No Capítulo 3 a abordagem utilizada é detalhada desde a detecção de pontos de interesse até a classificação dos vídeos. No Capítulo 4, apresentamos a base de vídeos criada para validação experimental, bem como os resultados obtidos em diferentes configurações experimentais. Por fim, o Capítulo 5 traz as conclusões, análises das contribuições e limitações do trabalho, além das direções para sua continuação.

Capítulo 2

Trabalhos relacionados

Os trabalhos desenvolvidos para reconhecimento de línguas de sinais usam geralmente ferramentas de reconhecimento de padrões e Visão Computacional. A área de Visão Computacional considera algoritmos de Processamento de Imagens e Computação Gráfica, mas se distingue por ter um propósito bem específico. A área de Processamento de Imagens trata de produzir uma imagem melhorada a partir de uma imagem de entrada. A área de computação gráfica trata de produzir uma imagem a partir de um modelo matemático do ambiente. Por sua vez, a Visão Computacional trata de interpretar e reconstruir um modelo do ambiente a partir de imagens de entrada. A Visão Computacional, portanto, estuda algoritmos para modelar objetos e reconhecer ações no ambiente usando imagens ou sequências de vídeo como entrada. Para tanto, utiliza técnicas de Inteligência Artificial, como aprendizado de máquina e reconhecimento de padrões.

No contexto do reconhecimentos de objetos, uma ou mais imagens podem ser usadas para extrair características que permitam construir um modelo para o reconhecimento do objeto. Nesse caso, métodos lineares, tais como Support Vector Machines (SVM) [Liu et al., 2016; Almeida et al., 2014] e LDA [Yan et al., 2014; Iosifidis et al., 2015] são amplamente usados. No caso do reconhecimento de ações, geralmente o uso de imagens estáticas limita o espectro de ações que podem ser reconhecidas, pois a dinâmica dos movimentos não pode ser capturada em imagens isoladas. Por isso, frequentemente, sequências de vídeo são usadas para construir modelos de reconhecimento de ações em geral. Então, trabalhos que envolvem o reconhecimento de ações utilizam amplamente métodos de classificação paramétricos, que capturam a dinâmica, como o Hidden Markov Models (HMM) [Zhao et al., 2017; de Souza & Pizzolato, 2013]. Por outro lado, sistemas de classificação com abordagens bioinspiradas têm atenção crescente em diversos campos [Xue et al., 2016; Huerta et al., 2010; Ribeiro et al.,

2015], mas ainda não são muito exploradas no contexto do reconhecimento de gestos. Este trabalho, então, explora uma abordagem baseada na combinação do ClonALG [de Castro & Zuben, 2002] com o algoritmo de treinamento da rede neural de Kohonen [Kohonen, 2001] para reconhecimento da Libras.

Nas seções seguintes, contextualizamos alguns trabalhos relacionados ao reconhecimento de ações em geral, o reconhecimento de línguas de sinais no mundo e, finalmente, trabalhos sobre reconhecimento da Língua Brasileira de Sinais.

2.1 Reconhecimento de ações

O reconhecimento de ações é amplamente investigado devido à variedade de aplicações que demandam essa tecnologia, à facilidade de acesso às câmeras digitais e a quantidade de vídeos produzidos no cotidiano da sociedade atual. Além disso, essa área desperta interesse da comunidade acadêmica, que explora técnicas de Visão Computacional para o reconhecimento de ações nos mais diversos contextos. Exemplos disso são os sistemas de realidade virtual, sistemas de controle e automação industrial, interfaces avançadas, jogos, etc.

Uma pesquisa recente apresentada por [Subetha & Chitrakala, 2016], resume uma variedade de metodologias e questões sobre sistemas de reconhecimento de ação humana e explora vários conjuntos de dados e suas propriedades. Na mesma direção, [Dawn & Shaikh, 2016] apresenta uma revisão do reconhecimento de ações humanas em vídeo, onde várias técnicas são organizadas de acordo com as etapas de detecção de pontos de interesse, descritores de características, construtores de vocabulário e classificadores. Eles também resumem os conjuntos de dados de vídeos públicos úteis para comparar desempenho de tais técnicas.

A maioria dos métodos, entretanto, tem várias limitações sobre as condições de aquisição de vídeos, tais como pose, iluminação, resolução, fundo e oclusão. Na direção oposta, [Idrees et al., 2017] e [Patel et al., 2016] propõem reconhecimento de ação em vídeos sem restrições. Enquanto em [Idrees et al., 2017] é descrito uma base de referência com desafios e protocolos de avaliação para quantificar os resultados da classificação, [Patel et al., 2016] propõe um modelo de fusão, em que várias modalidades, características ou classificadores são combinados para classificar sequências de vídeo sem restrições.

2.2 Reconhecimento de línguas de sinais

Diferentemente do reconhecimento de ação humana em geral [Liu et al., 2016; Zhao et al., 2017], o reconhecimento de línguas de sinais é particularmente desafiador e tem suas próprias restrições. Uma delas é que as línguas de sinais se diferem em cada país.

Alguns dos trabalhos pesquisados, envolvendo línguas de sinais em diversos países, usam tanto abordagens baseadas apenas em algoritmos de Visão Computacional, com uso de imagens e vídeos, como algoritmos que dependem de dispositivos mais sofisticados como luvas adaptadas ou outros sensores. Tanto o uso de imagem e vídeo como o uso de sensores possuem vantagens e desvantagens quando comparadas entre si. Por exemplo, a necessidade de o usuário usar luvas adaptadas ou sensores para interagir com o sistema tradutor gera uma diminuição na usabilidade da interação humano-computador, porém apresenta um grande ganho quanto à precisão na captação de movimentos [Kuroda et al., 2004]. Em alguns casos, luvas são usadas adaptadas com cores diferentes para os dedos, ou cores diferentes para dorso e palma da mão, com o objetivo de facilitar o processo de segmentação. Outra forma é a incorporação de sensores nas articulações da luva, o que permite o cálculo preciso de ângulos e, consequentemente, a inferência da configuração de mão. Já as características da abordagem baseada em Visão Computacional permitem uma interação mais intuitiva com a máquina, pois não há a necessidade de vestir nenhum tipo de equipamento [Yang, 2010].

Em [Kuroda et al., 2004], é exposto um exemplo em que, usando luvas, o método obtém grande ganho na captura de movimento. Em outros trabalhos, como em [Anetha & Parvin, 2014], luvas são combinadas com sensores para capturar os gestos, tornando o acesso ainda mais difícil para o público em geral. Dentre os trabalhos que utilizam Visão Computacional para reconhecimento de línguas de sinais, sem utilização de luvas ou sensores, destaca-se a segmentação e rastreamento de mão, geralmente usando HMM, que são estruturas amplamente utilizadas na modelagem de problemas com variações temporais. A Figura 2.1 ilustra como essa abordagem é aplicada em imagens amostradas de vídeos contendo sinais.

Quando se trata do reconhecimento de uma quantidade muito grande de itens, como é o caso das línguas de sinais, as técnicas de reconhecimento que constroem um classificador por gesto são inadequadas, conforme indicado em [Cooper et al., 2012]. Uma abordagem emergente para solução desse problema é o reconhecimento de subcomponentes do sinal. Já [Bowden et al., 2004] e [Kadir et al., 2004] apresentam trabalhos com modelos de classificação baseados em definições linguísticas dos sinais e utilizam

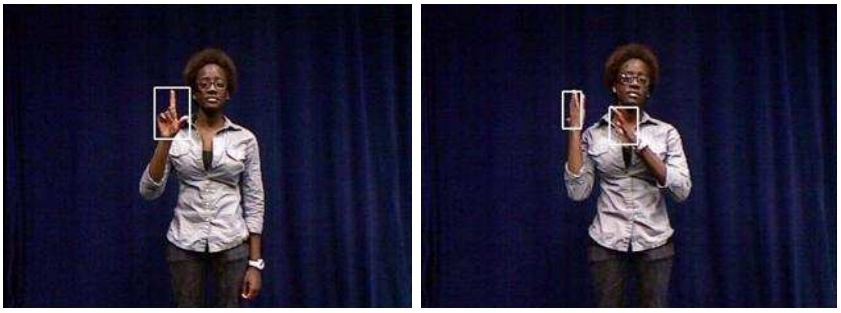


Figura 2.1. Amostras de quadros de vídeos contendo Língua Americana de Sinais. Imagens extraídas de [Jangyodsuk et al., 2014]

Modelos de Markov de 1ª Ordem (M1O), permitindo que sinais sejam aprendidos de forma confiável com pouco treinamento. O trabalho em [Cooper et al., 2012] segue a mesma linha, indicando que isso permite a escalabilidade do sistema.

Outros trabalhos combinam técnicas de segmentação para se adaptar melhor a ambientes complexos, com iluminação não controlada e variedade de texturas e cores [Chen et al., 2003; Caridakis et al., 2008]. Em [Neto & Oquendo, 2013], é apresentado um estudo do estado da arte das técnicas de reconhecimento de línguas de sinais por computador. Além disso, os diversos aspectos desse tipo de língua e seus desafios para o reconhecimento automático são discutidos, considerando as diversas fases do processo de implementação de um método para reconhecimento automático.

Devido à complexidade para realizar essa tarefa utilizando sequências de vídeo, estudos mais recentes utilizam outros recursos como mapas de profundidade ou dados RGB-D. No trabalho de [Dong et al., 2015] os dados RGB-D são usados para descrever um método para o reconhecimento do alfabeto da Língua de Sinais Americana (ASL) usando uma câmera de profundidade para localizar a posição da articulação manual e obter a configuração da mão. Na mesma direção, [Wang et al., 2016] propõem um método para o reconhecimento da Língua de Sinais Chinesa (CSL), usando dados RGB-D, onde cada sinal é caracterizado em termos das posturas típicas das mãos. Um classificador de nível de sinal, que utiliza *Binary Patterns* (BPs), para a Língua de Sinais Britânica (BSL) é apresentado em [Cooper et al., 2012], onde uma câmera de profundidade também é usada para rastreamento 3D. Além disso, muitos desses trabalhos se limitam ao reconhecimento de letras do alfabeto representadas de forma manual nas respectivas línguas de sinais de cada país. Isso torna o estudo bastante limitado em relação a quantidade de sinais que essas línguas possuem.

Na Tabela 2.1, são apresentados alguns dos trabalhos de reconhecimento de línguas de sinais no mundo, bem como os métodos utilizados. Todos os trabalhos apresentados, que utilizam Visão Computacional para realizar o reconhecimento de línguas

de sinais, trabalham com imagens geralmente extraídas de um conjunto de vídeos que contêm os sinais. Alguns deles utilizam luvas coloridas para facilitar o processo de segmentação, que extrai da imagem as regiões de interesse e diminui a quantidade de dados a ser trabalhada. Outros utilizam sensores 3D, com informações sobre as coordenadas das articulações do esqueleto e mapa de profundidade, para facilitar a detecção das mãos e movimentos.

Em [Carneiro et al., 2009; Anetha & Parvin, 2014; Bastos et al., 2015] são utilizadas redes neurais (NN) para realizar a classificação. Já em [Pavan & Modesto, 2010], é utilizada uma Cascata de Classificadores (CC), que é uma funcionalidade da biblioteca *OpenCV* responsável por identificar vários tipos de objetos em uma imagem. Um casamento de modelos é utilizado em [da Silva Júnior, 2014], em dois passos: (i) *casamento* da imagem de teste com um conjunto de modelos representativos; e (ii) *comparação* pareada destas imagens para estimar o grau de proximidade em cada par. Assim, modelos de referência que provejam melhores similaridades, para uma determinada métrica de correspondência, são utilizados para identificar a classe à qual o dado de teste pertence.

Trabalhos mais recentes de reconhecimento de línguas de sinais, usam algoritmos de Visão Computacional para extrair características e fazem a classificação com classificadores usuais da literatura. No trabalho de [de Paula Neto et al., 2015], por exemplo, é proposto um método que utiliza *Extreme Learning Machine* (ELM), enquanto que [dos Santos, 2015] trabalha com *K - Nearest Neighbors* (KNN) e [Dong et al., 2015] usa *Random Forest* (RF). O trabalho de [Wang et al., 2016] usa *Stable Marriage Problem* (SMP) e [Carneiro et al., 2016] utiliza *Eigenhands* com Distância Euclidiana (EDE). Amplamente usado no reconhecimento de objetos em imagens, a abordagem *Bag of Visual Features* (BoVFs) é também usada no reconhecimento de gestos, como em [de Souza, 2014; Cardenas & Chavez, 2015]. Essa abordagem é utilizada também em nosso trabalho na etapa de construção dos descritores para as sequências de vídeo.

2.3 Reconhecimento de Língua Brasileira de Sinais (Libras)

O reconhecimento de sinais da Libras é estudado principalmente em relação ao alfabeto manual, identificando as letras da LP através das configurações de mãos [Bragatto et al., 2009; Carneiro et al., 2009; Pavan & Modesto, 2010; da Silva Júnior, 2014; dos Santos, 2015; Carneiro et al., 2016]. No entanto, trata-se de um pequeno escopo

Língua/País	Referência	Luvas	Sensor 3D	Classificação
Taiwan	[Chen et al., 2003]	X		HMM
Inglaterra	[Bowden et al., 2004]			M1O
Inglaterra	[Kadir et al., 2004]			M1O
Japão	[Kuroda et al., 2004]	X		-
Brasil	[Marcotti et al., 2007]			DT
Brasil	[Souza et al., 2007]			HMM
Grécia	[Caridakis et al., 2008]			HMM
EUA	[Bragatto et al., 2009]	X		SVM
Brasil	[Carneiro et al., 2009]			NN
Brasil	[Pavan & Modesto, 2010]			CC
China	[Yang, 2010]			SVM
Brasil	[Siola, 2010]	X		HMM
Inglaterra	[Cooper et al., 2012]		X	HMM e BPs
Brasil	[de Souza & Pizzolato, 2013]		X	SVM
EUA	[Anetha & Parvin, 2014]	X		NN
EUA e Brasil	[da Silva Júnior, 2014]		X	CM
Brasil	[Almeida et al., 2014]		X	SVM
Brasil	[de Souza, 2014]		X	BoVFs
Brasil	[de Paula Neto et al., 2015]			ELM
Brasil	[Bastos et al., 2015]			NN
Brasil	[Cardenas & Chavez, 2015]		X	BoVFs
Brasil	[dos Santos, 2015]		X	KNN
EUA	[Dong et al., 2015]	X		RF
China	[Wang et al., 2016]		X	SMP
Brasil	[Carneiro et al., 2016]		X	EDE

Tabela 2.1. Trabalhos de reconhecimento de línguas de sinais de vários países.

em relação ao vocabulário da Libras que encontramos registrado no Novo Deit-Libras [Capovilla et al., 2013], que engloba 9.828 sinais.

Algumas iniciativas no sentido de reconhecer sinais da Libras, para além do alfabeto manual, são encontradas em [Marcotti et al., 2007; Souza et al., 2007; Siola, 2010]. Em [Marcotti et al., 2007], é desenvolvido um trabalho com fotos de letras e sinais. Essas imagens são pré-processadas manualmente para facilitar a subtração do fundo e a segmentação das mãos e, para classificação, é utilizada uma *Decision Tree* (DT).

Em [Souza et al., 2007], é utilizado um banco de imagens. Uma parte possui fundos estáticos e uniformes e outra com fundos dinâmicos em ambientes diversificados. São utilizadas imagens de 4 usuários, dois homens e duas mulheres, sendo que cada um executou três vezes um conjunto de 50 sinais selecionados, produzindo 12 amostras para cada sinal, gerando um total de 600 amostras. Durante a análise são extraídas,

manualmente, 11 características, de aproximadamente 60.000 imagens contidas nas 600 amostras de vídeos. O reconhecimento dos sinais é realizado utilizando HMM.

Uma outra base de vídeos de sinais da Libras é desenvolvida em [Siola, 2010] para o treinamento do módulo de reconhecimento proposto. São definidos 50 sinais distintos para reconhecimento. Para cada um desses sinais é realizada a gravação de 20 vídeos. Com o intuito de gerar uma sistema de reconhecimento independente de usuário, os vídeos são gravados por 2 intérpretes de Libras distintos, sendo um destes uma mulher, com conhecimento avançado em Libras e o outro um homem sem conhecimento prévio da língua. O módulo de reconhecimento de sinais criado e utilizado nesse trabalho é baseado em HMM e os sujeitos filmados usaram luvas, sendo uma azul na mão direita e uma laranja na mão esquerda, para facilitar o processo de segmentação.

Em estudos mais recentes, a pesquisa segue a mesma tendência das demais línguas de sinais, utilizando informações 3D a partir de sensores de profundidade. Por exemplo, [Almeida et al., 2014] apresenta um método de extração de características que explora a estrutura de Libras utilizando o sensor RGB-D para obtenção de dados de profundidade, de forma que eles investigam a relação entre características e elementos estruturais com base na forma, movimento e posição da mão. Em [Cardenas & Chavez, 2015] os autores descrevem um método para reconhecer Libras utilizando sensor de profundidade para obter esqueleto articulado e propor uma abordagem dinâmica de reconhecimento gestual. Na mesma direção, [Carneiro et al., 2016] usam esqueletos do sensor 3D para propor uma aplicação para geração de voz baseada em gestos de mão estáticos para usuários de Libras.

Já em [de Souza, 2014] é desenvolvido um trabalho voltado para reconhecimento de sinais da Libras com vistas ao desenvolvimento de jogos educacionais para alfabetização de surdos. Nele, é utilizado o sensor Microsoft Kinect para capturar nove sinais. Os vídeos são processados de forma a gerar um histograma por vídeo, numa abordagem BoVFs.

Outros trabalhos sobre o reconhecimento de Libras como [de Paula Neto et al., 2015], [Bastos et al., 2015] e [de Souza & Pizzolato, 2013] não usam sensores de profundidade. No entanto, eles são limitados a gestos de mão estática para reconhecer letras simples do alfabeto, números e uma quantidade muito pequena de palavras.

A maioria das bases para classificação de Libras encontradas na literatura são limitadas em quantidade e níveis de dificuldade para classificação de sinais. A Tabela 2.2 apresenta algumas bases de dados criadas para o estudo de reconhecimento de sinais da Libras.

As bases para realização de experimentos que visem o reconhecimento de sinais, e não apenas letras ou outras configurações de mãos, são limitadas. Além disso, esses

Referência	Sinais	Pessoas	Vídeos	Imagens	Sensor 3D
[Marcotti et al., 2007]	21	-		X	
[Souza et al., 2007]	50	4		X	
[Carneiro et al., 2009]	26	3		X	
[da Silva Júnior, 2014]	26	1		X	X
[de Souza, 2014]	9	23	181	X	X
[Bastos et al., 2015]	40	5		X	
[dos Santos, 2015]	61	10		X	X
[Carneiro et al., 2016]	20	4		X	X
base proposta	100	5	1500		

Tabela 2.2. Bases de dados de Libras.

trabalhos utilizam imagens para extração de características, não considerando a dinâmica temporal dos sinais. O presente trabalho, por outro lado, propõe uma base com 100 diferentes sinais em 1500 vídeos, com variados níveis de dificuldade para classificação. Isso devido aos diferentes níveis de variação de configurações, localização e movimento das mãos. Além disso, a base proposta utiliza apenas vídeos como entrada para o método de classificação. Essa base será melhor descrita na seção 4.1.

Capítulo 3

Fundamentação metodológica

A abordagem proposta para este trabalho consiste na classificação de sinais da Libras provenientes de vídeos com base na extração de características espaço-temporais, criando um vocabulário visual a ser usado na construção de descritores para cada vídeo. Essa abordagem, conhecida como *Bag Of Visual Features* (BoVFs), pode ser detalhada nos seguintes passos:

- Detecção de pontos de interesse;
- Extração de características;
- Construção do vocabulário visual;
- Construção de descritores para vídeos.

Enquanto que a *detecção de pontos de interesse* percorre cada pixel de cada quadro do vídeo para encontrar pontos cuja vizinhança tenha informação relevante, a *extração de características* codifica esses pontos como palavras visuais para *construção do vocabulário visual*. Esse vocabulário é usado para *construção de descritores* que serão usados para *classificação dos vídeos*. A Figura 3.1 ilustra a sequência de passos até a classificação com abordagem utilizada. Esses passos serão detalhados nas seções a seguir.

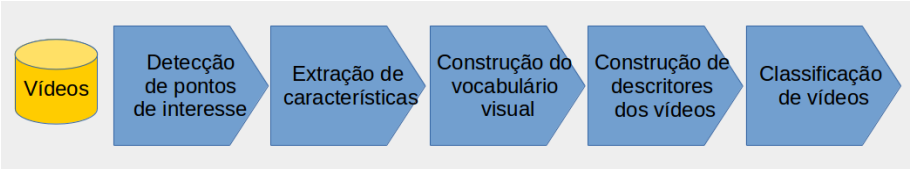


Figura 3.1. Passos da abordagem proposta para este trabalho

3.1 Detecção de pontos de interesse

São amplamente explorados algoritmos para detectar e descrever pontos de interesse, especialmente em imagens. Entre esses, os descritores *Speeded-Up Robust Features* (SURF) [Bay et al., 2008] e o *Scale Invariant Feature Transform* (SIFT) [Lowe, 2004] que se aplicam bem às imagens, são amplamente utilizados, estando implementados em diversas bibliotecas como *OpenCV*. A Figura 3.2 ilustra como o SIFT detecta relevantes variações espaciais em uma imagem, os chamados *pontos de interesse*. Com o conjunto de pontos de interesse de diversas imagens, os pontos similares são agrupados. Cada grupo, ou *cluster*, representa uma palavra visual do dicionário. Assim, uma imagem qualquer representada pela frequência de palavras visuais do dicionário pode ser comparada com outra para identificação do objeto. Com certo grau de certeza, é possível dizer se duas imagens correspondem ao mesmo objeto comparando os histogramas que as descrevem.

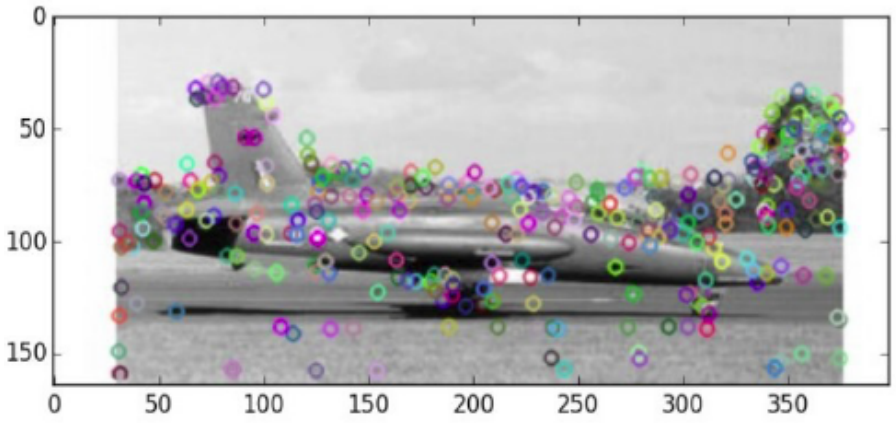


Figura 3.2. Exemplo de detecção de pontos de interesse em imagens. A imagem foi gerada a partir de uma implementação do SIFT utilizando a biblioteca *OpenCV*.

Esses algoritmos, entretanto, são limitados para extração de informações relevantes em vídeos, onde a dinâmica temporal é uma variável importante. Então, algoritmos do tipo *Space-Time Interest Points* (STIP) [Laptev, 2005; Willems et al., 2008] foram desenvolvidos para descrever pontos de interesse em vídeos, levando em conta as variações espaciais e temporais. Isso é necessário devido às estruturas da imagem em vídeo não se restringirem à velocidade constante e/ou aparecimento constante ao longo do tempo. Muitos eventos interessantes em vídeo são caracterizados por fortes variações nos dados ao longo de ambas as dimensões espacial e temporal. A Figura 3.3 ilustra a detecção de pontos de interesse em vídeos, usando o STIP.

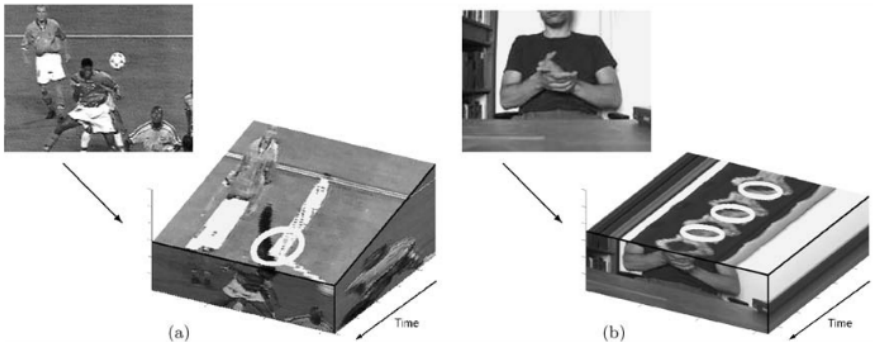


Figura 3.3. Exemplo de detecção de pontos de interesse em vídeos. Imagem extraída de [Laptev, 2005].

Neste trabalho, o algoritmo utilizado para a etapa de detecção dos pontos de interesse é baseado no detector Harris-Laplace, conforme proposto por [Laptev, 2005]. O algoritmo, denominado STIP, considera, nos domínios espacial e temporal, que o vídeo pode ser modelado pela sua representação espaço-escala linear definida pela convolução da sequência de vídeo com kernels gaussianos. Assim, o detector de pontos de interesse busca por locais no espaço de representação do vídeo com variações significativas em relação ao espaço e tempo. Considerando uma escala de observação, são detectadas as orientações de cada ponto em uma vizinhança local para construir uma matriz de covariância da distribuição dessas orientações para caracterizar a intensidade das variações locais. Essa caracterização é obtida considerando que os autovalores $\lambda_1, \lambda_2, \lambda_3$, em ordem crescente, da matriz de covariância indicam a variações ao longo das direções principais no espaço das escalas. Em particular, se todos todos os autovalores forem relativamente grandes, considerando um limiar, o ponto é considerado *ponto de interesse* [Laptev, 2005].

Esse algoritmo estende para vídeos a noção de pontos de interesse, inicialmente proposta para imagens, ao buscar por pontos numa vizinhança espaço-temporal que tenham significativas variações em todas as direções. Esses pontos, que representam locais no vídeo com ocorrência de movimento significativo, são considerados interessantes para o reconhecimento de ações em vídeos.

Para identificar esses pontos, o vídeo é considerado como um volume espaço-temporal definido por uma sequência de imagens, e representado por uma função escalar $f : \mathbb{R}^2 \times \mathbb{R} \longrightarrow \mathbb{R}$, que associa cada ponto (x, y, t) do vídeo a um escalar que indica intensidade. Então, pela convolução de f com um kernel gaussiano anisotrópico g de variância espacial independente σ_l^2 e variância temporal τ_l^2, é construída uma

representação escala-espacial $L : \mathbb{R}^2 \times \mathbb{R} \times \mathbb{R}^2_+ \mapsto \mathbb{R}$, dada por

$$L(\cdot\,; \sigma_l^2, \tau_l^2) = g(\cdot\,; \sigma_l^2, \tau_l^2) * f(\cdot\,), \tag{3.1}$$

onde o kernel gaussiano espaço-temporal separável é definido como

$$g(x, y, t; \sigma_l^2, \tau_l^2) = \frac{1}{\sqrt{(2\pi)^3 \sigma_l^4 \tau_l^2}} \times e^{\left(-\frac{x^2+y^2}{2\sigma_l^2} - \frac{t^2}{2\tau_l^2}\right)}. \tag{3.2}$$

No domínio espaço temporal, [Laptev, 2005] considera uma matriz μ (3×3) composta de derivativas médias espaciais e temporais de primeira ordem usando uma função de peso gaussiana $g(\cdot\,; \sigma_i^2, \tau_i^2)$ e dada por

$$\mu = g(\cdot\,; \sigma_i^2, \tau_i^2) * \begin{bmatrix} L_x^2 & L_x L_y & L_x L_t \\ L_x L_y & L_y^2 & L_y L_t \\ L_x L_t & L_y L_t & L_t^2 \end{bmatrix}, \tag{3.3}$$

onde as derivativas de primeira ordem L_x, L_y e L_t são definidas como

$$\begin{cases} L_x(\cdot\,; \sigma_l^2, \tau_l^2) = \partial_x (g * f) \\ L_y(\cdot\,; \sigma_l^2, \tau_l^2) = \partial_y (g * f) \\ L_t(\cdot\,; \sigma_l^2, \tau_l^2) = \partial_t (g * f) \end{cases} \tag{3.4}$$

e as escalas de σ_i^2 e τ_i^2 são relacionadas com as escalas locais σ_l^2 e τ_l^2, por uma constante s, de acordo com $\sigma_i^2 = s\sigma_l^2$ e $\tau_i^2 = s\tau_l^2$.

Então, essa matriz μ, construída para cada ponto do domínio de f, que representa o vídeo, é usada para filtrar aqueles que serão considerados *pontos de interesse*. Para tanto, são consideradas regiões de f onde μ tenha autovalores λ_1, λ_2, λ_3 significativos. A combinação desses autovalores permite distinguir regiões no vídeo com variação em uma, duas ou três dimensões, de acordo com a similaridade entre os autovalores. Em particular, regiões onde os três autovalores são similares e de grande intensidade indicam pontos com variação nas três dimensões e, portanto, de interesse espaço temporal [Laptev, 2005]

Essa combinação dos autovalores de μ, que privilegia a similaridade para detectar regiões com grande variação espacial e temporal é modelada pela combinação de determinantes e traços de μ, seguindo [Harris & Stephens, 1988], na seguinte expressão:

$$H = det(\mu) - k traco^3(\mu) = \lambda_1 \lambda_2 \lambda_3 - k(\lambda_1 + \lambda_2 + \lambda_3)^3. \tag{3.5}$$

Considerando valores suficientemente grandes para a constante k, o máximo local positivo de H corresponde a pontos no espaço-tempo com altos valores para λ_1, λ_2,

λ_3 e, desta forma, um limiar pode ser usado para filtrar aqueles que serão considerados de interesse por indicar estruturas do vídeo com características espaço-temporais importantes [Laptev, 2005].

A Figura 3.4 contém um esquema que resume essa etapa. Os vídeos são a entrada para execução do detector, que retorna como saída um conjunto dos pontos de interesse de cada vídeo. O módulo de detecção de pontos de interesse utilizado é o proposto em [Laptev, 2005].

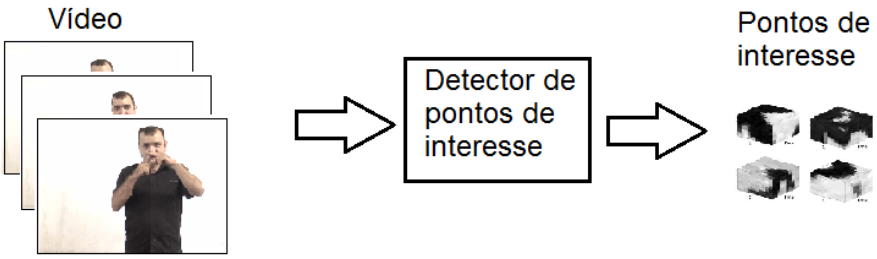


Figura 3.4. Esquema de detecção de pontos de interesse.

3.2 Extração de características

A extração de características busca descrever matematicamente os pontos de interesse detectados. Para isso, após serem identificados os pontos de interesse, são criados vetores descritores de cada um desses pontos. Esses descritores são utilizados para caracterizar os vídeos pela distribuição dos pontos de interesse, viabilizando o processo de comparação.

Para construção dos descritores locais para pontos de interesse detectados, a mesma função $f : \mathbb{R}^2 \times \mathbb{R} \longrightarrow \mathbb{R}$, usada para representar o vídeo na etapa de detecção, é considerada. Para tanto, são consideradas combinações de derivativas gaussianas computadas nas mesmas escalas espaço-temporais (σ, σ, τ) utilizadas para detectar os pontos de interesse, conforme proposto por [Laptev, 2005]. Essas combinações de derivativas gaussianas são definidas como:

$$L_{x^m y^m t^k} = \sigma^{m+n} \tau^k (\partial_{x^m y^n t^k} g) * f \tag{3.6}$$

Usando essas combinações, os descritores são definidos como derivativas locais até terceira ordem, denominados *jet* [Koenderink & van Doorn, 1992], calculados em escalas espaço-temporal, de forma que cada ponto de interesse é, então, descrito por

um vetor j de $n = 34$ componentes, dado por:

$$j = (L_x, L_y, L_t, L_{xx}, \ldots, L_{ttt}). \tag{3.7}$$

Com os descritores de pontos de interesse extraídos dos vídeos, um passo de redução de dimensionalidade dos descritores é executado para reduzir o custo computacional necessário nas etapas seguintes. Essa redução é feita, tipicamente, por meio do algoritmo de Análise de Componentes Principais (PCA).

Finalmente, considerando a matriz de covariância Σ relativa à distribuição dos pontos de interesse de um conjunto de treino, a comparação entre dois descritores j_1 e j_2 pode ser feita usando alguma métrica. Em [Laptev, 2005], a comparação considera a distância de Mahalanobis, dada por

$$d^2(j_1, j_2) = (j_1 - j_2)\Sigma^{-1}(j_1 - j_2)^T \tag{3.8}$$

A Figura 3.5 ilustra essa etapa, onde os pontos de interesse são entrada para o processo de extração de características. Supondo m pontos de interesse extraídos de um determinado vídeo, são então produzidos m descritores, cada um com n componentes. A saída do processo, para cada vídeo de entrada, é uma matriz $m \times n$ em que m é a quantidade de pontos de interesse detectados e n é a dimensão do vetor de descritores. Os melhores resultados obtidos nesse trabalho usam $m = 50$ e $n = 34$, conforme adaptação da implementação utilizada de [Laptev, 2005].

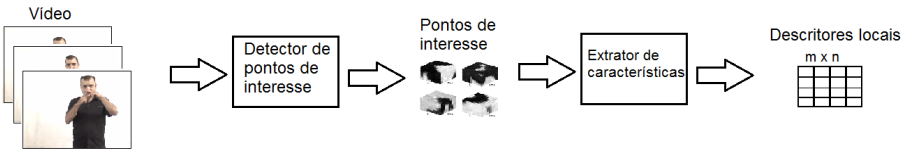


Figura 3.5. Esquema de extração de características.

3.3 Construção do vocabulário visual

Para representação dos vídeos, usando os pontos de interesse, será construído um vocabulário visual pelo agrupamento desses pontos por similaridade. Esse agrupamento de pontos de interesse similares leva ao conceito de *palavras visuais* que, em conjunto, definem um *vocabulário visual*. Nessa representação, conhecida como *Bag of Visual Features*, cada vídeo é representado por um histograma de distribuição das palavras visuais. Essa representação é inspirada na representação conhecida como *bag-of-word*

usada na classificação de documentos de texto. Conforme [Yang et al., 2007], essas representações são análogas em termos de forma e semântica, pois ambas são esparsas, de alta dimensionalidade e, da mesma forma que as palavras permitem caracterizar um documento, as palavras visuais caracterizam o padrão local de uma imagem.

Na construção do vocabulário visual, os descritores j dos pontos de interesse de todos os vídeos são agrupados por similaridade em um número de grupos, ou *clusters*, usando o algoritmo *K-means*. Conforme detalhado em [Yang et al., 2007], cada *cluster* determina uma palavra visual associada a um padrão local característico dos pontos de interesse pertencentes ao *cluster*. Então, os diferentes padrões locais são representados, no processo de agrupamento, pelas palavras visuais. O tamanho do vocabulário pode ser um parâmetro em cada aplicação, geralmente escolhido empiricamente, de acordo com a natureza dos dados.

Neste trabalho utilizamos o algoritmo de agrupamento *k-means* para o agrupamento. O *k-means* funciona basicamente em duas etapas. Dado um número inicial k, de grupos, e um conjunto de vetores descritores, são definidos k centróides e (i) cada descritor é associado ao centróide que minimiza a distância ao descritor, segundo alguma métrica, como Euclidiana por exemplo. (ii) Na segunda etapa, são calculados os novos centróides baseando-se na média de todos os descritores atribuídos ao grupo. O processo iterativo alterna entre a primeira e segunda etapas até que um critério de parada seja atingido. Um critério de parada pode ser o número de iterações ou um erro máximo acumulado pelas distâncias dos descritores ao centróide do seu grupo atual. Cada um dos centróides finais escolhidos é definido como uma palavra do vocabulário visual.

O algoritmo *k-means* está disponível na biblioteca *Scikit Learn* e alguns números aleatórios k de grupos e critérios de parada foram testados, verificando o que melhor contribui no desempenho de classificação. Os melhores resultados neste trabalho são obtidos para $k = 50$.

A Figura 3.6 traz um esquema que ilustra a formação do vocabulário visual a partir de um conjunto de descritores locais como entrada. Supondo que t vídeos distintos determinaram as t matrizes $m \times n$, conforme etapa anterior, obtemos por concatenação uma única estrutura de dados com tm vetores de dimensão n. Esses vetores são então agrupados em k grupos, utilizando o *k-means*. Como resultado, temos os centróides de cada um dos grupos gerados. Esses centróides, representam as k palavras visuais do vocabulário. O módulo para construção de vocabulário visual é desenvolvido, neste trabalho, em Python.

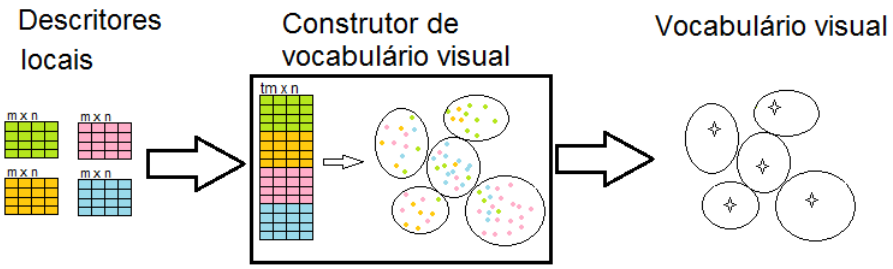


Figura 3.6. Esquema para construção de vocabulário visual.

3.4 Construção de descritores para vídeos

No caso de classificação de imagens, a representação *bag-of-word* pode ser convertida em um vetor de características que pode conter, em cada posição do vetor, a presença ou ausência de cada palavra visual na imagem [Yang et al., 2007]. Dessa forma, a distribuição de palavras visuais cria um histograma, onde cada posição tem a contagem do número de pontos de interesse associado a um agrupamento.

De maneira similar, neste trabalho, os descritores espaço-temporais dos vídeos são agrupados para determinar um vocabulário visual e depois é criado um vetor de características para cada vídeo. Isto é, para cada vídeo será construído um descritor que consta de um vetor de k posições para registrar o número de ocorrência de cada palavra visual no vídeo em questão. Cada ponto de interesse do vídeo é associado a uma palavra visual mais próxima no dicionário, num processo de contagem de palavras. Assim, o vídeo é representado pela distribuição de frequência de ocorrências das palavras visuais.

A Figura 3.7 ilustra um histograma para um dicionário de palavras visuais e a distribuição de frequência de ocorrências dessas palavras no vídeo.

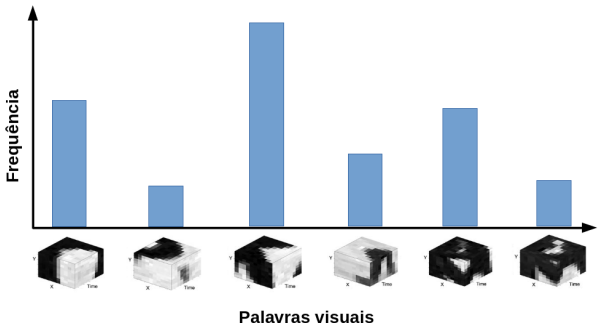


Figura 3.7. Ilustração do histograma de contagem de palavras visuais.

O módulo que constrói os descritores para os vídeos é desenvolvido em Python e recebe todos os vídeos como entrada. Com o vocabulário visual obtido na etapa anterior, são gerados descritores para cada vídeo, obtidos como vetores de tamanho k,

que representam histogramas de contagem de palavras visuais. A Figura 3.8 ilustra o esquema que resume essa etapa.

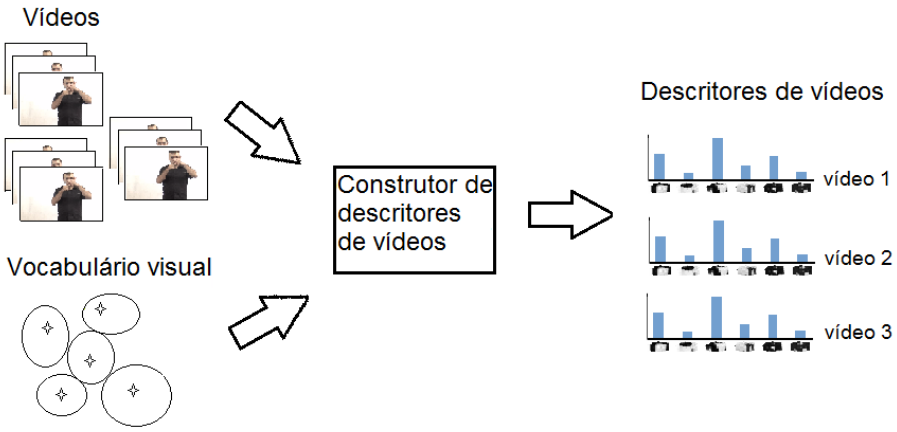


Figura 3.8. Esquema para construção de descritores de vídeos.

3.5 Classificação dos vídeos

Conforme etapa anterior de construção dos descritores, cada vídeo é representado para classificação por um vetor de $k = 50$ componentes. A etapa de classificação requer um processamento de alto nível que classifique os objetos de teste, através da comparação das características dos mesmos com aqueles objetos de treino que tem classes previamente estabelecidas.

No passo de classificação, as sequências de vídeo codificadas em termos de características visuais são usadas para construir um modelo ou função de reconhecimento. Tal modelo permite associar uma sequência de vídeo de teste não rotulada a uma classe previamente estabelecida, usando algoritmos de reconhecimento de padrões.

Nessa etapa, duas abordagens são consideradas. A primeira abordagem utiliza um classificador LDA [Yan et al., 2014; Iosifidis et al., 2015], disponível na biblioteca *Scikit Learn* . A segunda abordagem utiliza o algoritmo bioinspirado ClonALG [de Castro & Zuben, 2002], combinado com o algoritmo de treinamento da rede neural de Kohonen [Kohonen, 2001], a fim aumentar o desempenho de classificação, como apresentado em [D'Angelo et al., 2016]. No Anexo A, detalhamos a formulação matemática do classificador LDA, conforme disponível na biblioteca *Scikit Learn*. Na seção seguinte, a abordagem Imune/neural é detalhada:

3.5.1 Classificador Imune/neural

O classificador Imune/neural é baseado na combinação do ClonALG [de Castro & Zuben, 2002] com o algoritmo de treinamento da rede neural de Kohonen [Kohonen, 2001]. A abordagem ClonALG leva explicitamente em consideração a afinidade dos anticorpos onde apenas os mais aptos são selecionados para proliferar através de um processo chamado maturação (ou mutação). O critério de afinidade é determinado pelo método da distância euclidiana [D'Angelo et al., 2016].

Como em quase todos os algoritmos evolutivos, esta abordagem usa o conceito de populações e relações entre seus indivíduos para proliferar o mais apto. Nesse caso específico, temos uma população de antígenos e uma população de anticorpos que reconhecerão os antígenos. Ambas as populações podem ser vistas como matrizes (Equação 3.9), com cada linha correspondente a um indivíduo com dimensão m

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix} \in \mathbb{R}^{k \times m} \qquad \text{e} \qquad B = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \in \mathbb{R}^{n \times m}, \tag{3.9}$$

onde X é a população de antígenos e B a população de anticorpos; $x_i \in \mathbb{R}^m$ e $b_i \in \mathbb{R}^m$. A ideia geral de um algoritmo baseado em evolução é repetir os mecanismos de clonagem, mutação e seleção até que um critério de parada seja satisfeito. Esses passos são apresentados a seguir conforme [D'Angelo et al., 2016].

3.5.1.1 Critério de parada

A implementação utilizada neste trabalho considera dois critérios de parada. O primeiro é um simples limite ao número de gerações (uma iteração composta por clonagem, mutação e seleção). Esse número foi definido como 7, pois, após alguns testes, verificamos que eram obtidos resultados satisfatórios sem utilização de tempo excessivo. O segundo analisa a quantidade de anticorpos criados e eliminados em cada geração: se a quantidade de anticorpos eliminados for igual ou maior do que a quantidade de anticorpos criados, o algoritmo pode ser parado.

3.5.1.2 Clonagem

Cada anticorpo é clonado duas vezes e cada um destes clones sofre uma mutação aleatória, utilizando uma distribuição uniforme entre -1 e 1 na sua posição espacial.

Isso é mostrado na Equação 3.10 e garante a diversidade na população de anticorpos

$$B = \{B + \lambda\, \mathcal{U}_{n\times m}(-1,1)\} \;\cup\; \{B + \lambda\, \mathcal{U}_{n\times m}(-1,1)\} \;\cup\; B \tag{3.10}$$

onde $\mathcal{U}_{n\times m}(-1,1)$ representa uma matriz com $n \times m$ variáveis aleatórias uniformemente distribuídas entre -1 e 1 e λ é uma constante que será escolhida em relação aos dados. Ela pode ser $\max(B)$ ou mesmo $\text{mean}(B)$; o tipo de aplicação define o que melhor se encaixa.

3.5.1.3 Mutação

A mutação começa no mecanismo de clonagem e é melhorada por uma rede neural de Kohonen, onde cada antígeno reduz linearmente a distância entre ele (x_j) e o anticorpo mais próximo($\hat{b}_{x_j}$ dado pela equação 3.11). Este algoritmo altera a posição do anticorpo como se fossem pesos de um neurônio como mostrado na Equação 3.12.

$$\hat{b}_{x_j} = \underset{b_i}{\text{argmin}} \{\|x_j - b_i\| : b_i \in B\} \tag{3.11}$$

$$\hat{b}_{x_j} = \hat{b}_{x_j} + \alpha_k(x_j - \hat{b}_{x_j}) \tag{3.12}$$

$$\alpha_k = \alpha_{k-1}\left(1 - \frac{k}{K_m}\right) \tag{3.13}$$

Observamos que as Equações 3.11–3.13 iteram sobre k com $0 < k \leq K_m$ para todo $x_j \in X$. Para que essa redução linear funcione, devemos definir α_0 e K_m; neste trabalho, foi definido $\alpha_0 = 0.8$ e $K_m = 2$.

3.5.1.4 Seleção

A seleção usa duas informações para podar os anticorpos. A primeira é o tipo de antígenos reconhecidos principalmente por um anticorpo, dado pela Equação 3.14. A segunda usa a proximidade entre dois anticorpos, onde o limiar para considerar um anticorpo próximo a outro é definido na Equação 3.15.

$$r(b_i) = \text{Mo}\left(\{t(x_j) : x_j \in X \text{ and } \hat{b}_{x_j} = b_i\}\right) \tag{3.14}$$

onde $t(x_j)$ representa a classe do antígeno x_j e Mo é o modo do conjunto.

$$c = 0.25\left(\frac{2}{3n(3n-1)} \sum_{i=0}^{3n-1} \sum_{j=i+1}^{3n} \|b_i - b_j\|\right) \tag{3.15}$$

Em outras palavras, o limiar é de 25% da distância média entre cada dois anticorpos. Iniciamos pela remoção de anticorpos que não reconhecem qualquer classe de antígeno (Equação 3.16)

$$B = \{\hat{b}_{x_j} : x_j \in X\} \cap B \tag{3.16}$$

Em seguida, analisamos todas as combinações de dois elementos fora do conjunto de anticorpos. Quando o conjunto de antígenos reconhecidos por dois deles são do mesmo tipo (Equação 3.14) e estão próximos uns dos outros (Equação 3.15), um novo anticorpo é criado entre eles e esses dois devem ser apagados (Equação 3.17).

$$B_1 = \left\{ \frac{b_i + b_j}{2} : (b_i, b_j) \in \binom{B}{2} \text{ and } r(b_i) = r(b_j) \text{ and } \|b_i - b_j\| < c \right\}$$

$$B_2 = \left\{ b_i; b_j : (b_i, b_j) \in \binom{B}{2} \text{ and } r(b_i) = r(b_j) \text{ and } \|b_i - b_j\| < c \right\}$$

$$B = (B \setminus B_2) \cup B_1 \tag{3.17}$$

Da Equação 3.17, é possível ver que esse método não garante que a população final de anticorpos seja de tamanho n.

Capítulo 4

Resultados Experimentais

Os experimentos realizados para validar a abordagem proposta neste trabalho de reconhecimento da Libras, além dos resultados, são apresentados e discutidos neste capítulo. No primeiro momento, foi detalhado o conjunto de dados de vídeo com 100 sinais diferentes usados nos experimentos. No segundo momento, alguns classificadores da biblioteca *Scikit-Learn* foram testados e o LDA, que obteve melhor desempenho, foi selecionado para os experimentos seguintes. Além disso, foram testados os parâmetros de quantidade de pontos de interesse detectados em cada vídeo (m) e o número (k) de palavras visuais usadas na construção do vocabulário visual. O conjunto de descritores dos vídeos, usando os parâmetros ajustados, foi utilizado para testes com a base particionada. No terceiro momento, o método de classificação Imune/neural foi comparado com o método de Análise de Discriminante Linear, utilizando a base de dados de descritores de vídeos. A escalabilidade de ambos os classificadores foi explorada para subconjuntos crescentes de sinais, de 2 a 100 sinais diferentes.

Os testes foram executados em um computador com processador Intel Core i5 com 3,5 GB de RAM. Os seguintes recursos de software foram utilizados para implementação da abordagem:

- Detecção de pontos de interesse e extração de características: foi utilizada uma implementação do STIP [Laptev, 2005] desenvolvida no Matlab;

- Construção do vocabulário visual e de descritores para vídeos: foi criada uma implementação própria em Python, sendo utilizado o *k-means* do *OpenCV*;

- Classificação dos vídeos: a primeira parte foi implementada em Python, utilizando classificadores do *Scikit-Learn*. A segunda parte foi implementada em

1 - água	26 - dó	51 - pediatra	76 - copo
2 - pênis	27 - saúde	52 - difícil	77 - beber
3 - lanche	28 - casa	53 - hospital	78 - comunicar
4 - ontem	29 - escola	54 - menstruação	79 - falhar
5 - ter	30 - igreja	55 - frio	80 - aquele
6 - atrasar	31 - açougue	56 - computador	81 - pessoa passando
7 - antes	32 - padaria	57 - sacanagem	82 - pessoa afastando
8 - queijo	33 - desculpa	58 - peixe	83 - pessoa aproximando
9 - rir	34 - perdoar	59 - escorpião	84 - interesse
10 - televisão	35 - boi	60 - papagaio	85 - alto
11 - trabalhar	36 - estratégia	61 - médico	86 - barulho
12 - solteiro	37 - gordo	62 - reunião	87 - eu
13 - sogro	38 - telefone	63 - pão	88 - aqui
14 - sábado	39 - divulgar	64 - arrepender	89 - fofoqueiro
15 - aprender	40 - avisar	65 - aluno	90 - biscoito
16 - amar	41 - informação	66 - imagem	91 - perigo
17 - ouvinte	42 - coração	67 - saudade	92 - eu vou
18 - desodorante	43 - febre	68 - agosto	93 - você vem
19 - namorar	44 - biblioteca	69 - banho	94 - encontrar
20 - Jesus	45 - banco	70 - cadeira de rodas	95 - diálogo
21 - não saber	46 - porta	71 - junto	96 - dinâmica
22 - não acreditar	47 - falso	72 - perto	97 - decreto
23 - saber muito	48 - amigo	73 - esporte	98 - deficiente
24 - fácil	49 - cardiologista	74 - quente	99 - direito
25 - amanhã	50 - obstetra	75 - comportamento	100 - regra

Tabela 4.1. 100 sinais que compõem a base de dados produzida.

Matlab, utilizando uma implementação do classificador Imune/neural conforme em [D'Angelo et al., 2016].

4.1 Criação da base de dados

Para validação da abordagem utilizada para reconhecimento de sinais da Libras, foi criada uma base de dados de vídeos de 100 sinais diferentes de Libras. Para isso, 4 sujeitos surdos nativos e um ouvinte fluente em Libras contribuíram para as gravações. Esses 5 sujeitos são usuários fluentes em Libras há mais de 10 anos, o que garantiu a criação de uma base confiável. Isso ocorre devido a naturalidade dos sinalizadores surdos e/ou ouvintes fluentes em Libras diferenciar-se da mera reprodução de gestos realizados por ouvintes que desconhecem a língua.

O vocabulário em Libras selecionado para os experimentos foi descrito na Tabela 4.1, em sua correspondência em LP, e numerado por diferentes ações.

Os sinais foram selecionados considerando aspectos linguísticos como a sinalização por diferentes sujeitos, as variações visuais dos sinais, além das marcas não manuais que fazem parte da língua e produzem significado. Alguns sinais são compostos de movimentos que são facilmente distinguíveis pela distinção entre configurações de mãos, localização e movimento. Outros sinais têm esses parâmetros semelhantes, tornando a classificação mais desafiadora. Visualmente, isso pode ser verificado nas imagens extraídas dos vídeos contendo sinais. A Figura 4.1 apresenta imagens para os sinais de *atrasar* e *antes*. Esses sinais possuem mesma configuração de mãos e posição, mas movimentos diferentes. A Figura 4.2 contém imagens dos sinais de *comunicar* e *dinâmica*. Esses sinais são similares quanto ao movimento, mas possuem diferentes configurações de mãos. A Figura 4.3 apresenta imagens dos sinais de *aprender* e *arrepender*. Ambos com mesma localização, mas diferentes configurações de mãos e movimento. Por fim, a Figura 4.4 com imagens dos sinais *amar* e *decreto*, ilustra dois sinais facilmente diferenciáveis.

a) Sinal de *atrasar* b) Sinal de *antes*

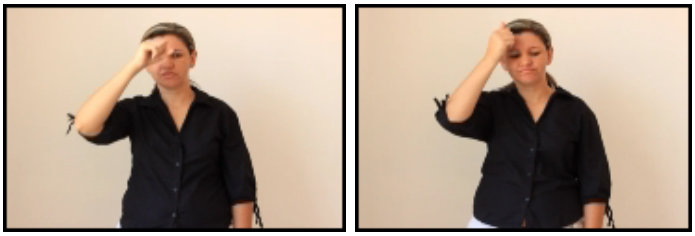
Figura 4.1. Sinais com configuração de mãos similar: (a) *atrasar* e (b) *antes*.

a) Sinal de *comunicar* b) Sinal de *dinâmica*

Figura 4.2. Sinais com movimento de mãos similar: (a) *comunicar* e (b) *dinâmica*.

A realização da filmagem foi feita por um profissional e as seguintes diretrizes foram respeitadas:

- Foram selecionados 100 sinais da Libras para compor a base de dados. Cada um dos sinais foi filmado três vezes por cada um dos 5 sujeitos, totalizando 1500 vídeos.



a) Sinal de *aprender* b) Sinal de *arrepender*

Figura 4.3. Sinais com localização similar: (a) *aprender* e (b) *arrepender*.

a) Sinal de *amar* b) Sinal de *decreto*

Figura 4.4. Sinais facilmente diferenciáveis: (a) *amar* e (b) *decreto*.

- O corte dos vídeos deu-se quando a pessoa parava com mãos para baixo, próximas ao corpo.
- A iluminação do ambiente foi controlada, evitando ao máximo o aparecimento de sombras nos vídeos.
- Os sujeitos usaram roupas pretas, buscando enfatizar o contraste com o tom de pele e cor de fundo.
- O fundo foi de cor uniforme, claro e estático.
- Os sujeitos não usaram acessórios como joias, relógios, bonés, etc; para não refletir a luz e/ou haver distração do objetivo alvo.
- A câmera foi fixada de frente para o sujeito. O enquadramento deixou espaço de aproximadamente 20cm nas laterais, acima da cabeça e abaixo da cintura.
- Um orientador executou os 100 sinais escolhidos de frente para cada sujeito, que o reproduziu à sua maneira três vezes (obedecendo as pausas para cortes entre sinais, como explicado anteriormente).
- Cada sujeito filmado assinou um termo permitindo que seus vídeos fossem utilizados para pesquisa científica.

Consideramos que a participação de surdos nativos ou ouvintes fluentes, com ampla convivência com surdos, tornou a base mais confiável, pois trata-se de uma língua complexa e não de mera gesticulação. A repetição dos sinais por cada pessoa também foi uma característica interessante da base, pois aumentou a quantidade de exemplos para serem usados nos treinamentos e testes. Além disso, os cuidados com contrastes de roupa e cor da pele, além da ausência de acessórios, são amplamente utilizados pelos intérpretes de Libras em ocasiões formais. Isso pelo fato de a língua ser visual, exigindo esforço desse sentido pelos interlocutores. Outro ponto considerado foi a posição da câmera e enquadramento, que também são amplamente utilizados em filmagens de Libras em contextos formais. Diante disso, julgamos que a base de dados criada seja adequada para o desenvolvimento de aplicações que utilizam a Visão Computacional para o reconhecimento de sinais da Libras em ambiente controlado.

Cada um dos 1500 vídeos tem duração média de 3 segundos e foram gravados com 30 quadros por segundo, de forma que cada vídeo tem, em média, 90 quadros. Originalmente os vídeos foram gravados no formato "*mp*4" em cores com resolução 720×480 pontos por quadro e, numa fase de pré-processamento, foram convertidos para o formato "*avi*", em escala cinza e com resolução de 180×120 pontos por quadro.

Para facilitar o planejamento dos experimentos, os vídeos receberam nomes que identificam a ação, o sujeito e o exemplo. Cada vídeo recebeu nome "aXsYeZoW.avi", onde:

- X indica o número da ação, de 1 a 100;
- Y indica o número do sujeito, de 1 a 5;
- Z indica o número do exemplo, de 1 a 3.

Em alguns vídeos, houve problemas de gravação e os mesmos não puderam ser usados nos experimentos, sendo substituídos por outros exemplos da mesma ação e mesmo sujeito. Os arquivos de vídeo com esses casos podem ser identificados quando Z é diferente de W, e totalizam 14 dos 1500 vídeos (menos de 1%).

4.2 1º Configuração experimental

Na primeira configuração experimental, alguns classificadores disponíveis na biblioteca *Scikit-Learn* foram testados. Os parâmetros de quantidade de pontos de interesse e palavras visuais foram ajustados para criação de um conjunto de descritores dos vídeos, de forma a maximizar os resultados da classificação. Além disso, testes com a base particionada em grupos de sinais, mais e menos desafiadores, foram realizados.

4.2.1 Classificadores Scikit-Learn

O primeiro teste objetivou a verificação do desempenho de alguns classificadores disponíveis na biblioteca *Scikit-Learn*, usando as configurações padrão. Inicialmente foram escolhidos 10 sinais com características linguísticas diferentes, visando facilitar a tarefa de classificação. Os sinais foram correspondentes em LP para: *perigo, padaria, interesse, pediatra, telefone, biscoito, peixe, aprender, computador* e *ouvinte*.

Para realização desse experimento, foram detectados os pontos de interesse e extraídas as características dos 150 vídeos, sendo 10 sinais diferentes e 15 vídeos exemplos por sinal. O resultado dessa etapa foi um conjunto de 150 arquivos de texto, cada um contendo os descritores para pontos de interesse espaço-temporal extraídos do vídeo.

A implementação utilizada para detecção de pontos de interesse e extração de característica de [Laptev, 2004] considera os pontos que possuem variação espacial e temporal segundo um limiar mínimo. Observando os arquivos com os descritores espaço-temporais gerados, contendo tais pontos de interesse de cada vídeo, verificamos que a quantidade de pontos de interesse detectados foi muito pequena. Esse número variou entre 1 e 10 pontos de interesse por vídeo, aproximadamente.

O próximo passo foi a execução do algoritmo de classificação, implementado em Python, que foi executado 30 vezes, gerando dados estatisticamente confiáveis para nossas análises. Em cada execução, uma amostra de cada um dos 5 sujeitos foi separada para teste, enquanto duas foram usadas para treinamento. Então, o conjunto de treinamento contou com 100 amostras e o conjunto de testes 50 amostras. Para tarefa de agrupamento em palavras visuais, definimos 50 grupos ($k = 50$). O algoritmo pode ser descrito pelos seguintes passos:

1. Separar os exemplos para o conjunto de teste;
2. Separar os exemplos para o conjunto de treinamento;
3. Realizar o agrupamento dos descritores espaço-temporais do conjunto de treinamento, formando o vocabulário visual;
4. Criar os descritores dos vídeos do conjunto de treino em termos do vocabulário visual;
5. Criar os descritores dos vídeos do conjunto de teste em termos do vocabulário visual;;
6. Treinar o classificador usando o conjunto de treinamento de forma a obter o modelo para classificação;

7. Classificar o conjunto de teste usando o modelo obtido.

Usando a biblioteca *Scikit Learn*, foram testados os classificadores LDA, SVM, Random Forest, KNN e Decision Tree, todos considerando a configuração padrão da biblioteca. Os resultados desse teste são apresentados na Tabela 4.2. Podemos verificar que o classificador LDA obteve os melhores resultados e, portanto, foi selecionado para ser utilizado nos experimentos seguintes.

Tabela 4.2. Acurácia dos classificadores para testes com 10 sinais com parâmetros linguísticos facilmente diferenciáveis.

Classificador	Acurácia (%)
LDA	65.26
SVM	64.00
Random Forest	61.40
KNN	57.86
Decision Tree	54.13

4.2.2 Teste com 100 sinais e com base particionada

Nesse teste, todos os 100 sinais foram utilizados para treinamento e teste. Foram 1400 vídeos para treinamento e 100 vídeos para os testes. De cada tipo de sinal, a base contou com 15 vídeos, dos quais um deles foi retirado para testes e os demais 14 usados para treinamento. Os experimentos foram executados 15 vezes, de forma que, a cada vez, um vídeo diferente foi destacado para teste e o restante para treino, numa estratégia *15-fold*. Como resultado da taxa de reconhecimento do classificador, foi calculada a média das 15 execuções e desvio padrão. Nesse experimento, com os 100 sinais e utilizando o LDA, a taxa média de reconhecimento geral foi de 36.67%. De fato, apesar da baixa taxa de classificação entre os 100 sinais, alguns obtiveram taxas muito altas e outros taxas muito baixas, comprometendo a média geral.

Observando que o nível de dificuldade aumenta com a quantidade de sinais, selecionamos dois grupos, de 25 sinais cada, para apresentar os resultados da base particionada. No Grupo 1, foram selecionados 25 sinais considerados de fácil classificação, por ter movimentos distintos e com pouca ambiguidade. Esses sinais foram os que obtiveram maior acurácia na classificação utilizando 100 sinais. No grupo 2, foram selecionados 25 sinais de difícil classificação, com movimento similares e alguma ambiguidade. Portanto, esses sinais foram os que obtiveram pior acurácia na classificação utilizando 100 sinais.

No experimento com os grupos 1 e 2, foi usado o mesmo particionamento entre conjunto de treinamento e teste, ou seja, a cada ação 14 vídeos foram usados para treinamento e 1 vídeo usado para teste. Assim, o conjunto de treino contou com 350 vídeos e o conjunto de teste com 25 vídeos. Os resultados foram apresentados na forma de médias de classificação com respectivos desvios. A Tabela 4.3 mostra as médias de acurácia e desvio padrão para cada um dos grupos.

Tabela 4.3. Acurácia e desvio padrão de classificação por grupos.

	Grupo 1 (Sinais distintos)	Grupo 2 (Sinais similares)
Acurácia	71.30%	43.20%
Desvio padrão	28.68	32.67

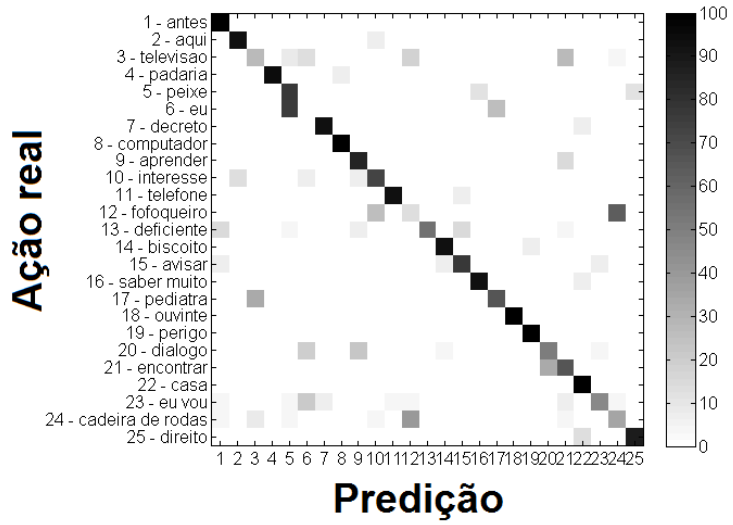
Observamos que, conforme previsto na organização dos grupos, a acurácia resultante do classificador LDA no grupo 1 foi superior ao grupo 2. A Figura 4.5, contém as matrizes de confusão geradas a partir da execução do algoritmo em cada grupo. Essa matriz indica, em escala de cinza, as taxas de acerto para cada sinal. Na matriz de confusão, células brancas indicam 0% de correspondência e células pretas indicam 100% de correspondência. Idealmente, a diagonal deveria ser preta com demais células brancas ao redor. Mas, como pode ser observado, as células escuras fora da diagonal indicam erros de classificação.

Apesar de que a classificação com sinais similares do grupo 2 continua um desafio a ser considerado, os resultados com o Grupo 1 mostraram que a abordagem utilizada foi capaz de classificar os sinais com taxa de reconhecimento superior a 70%, quando os sinais são bem distintos. Considerando o tamanho da base de dados e a complexidade da classificação em vídeo, os resultados indicaram que a abordagem é promissora.

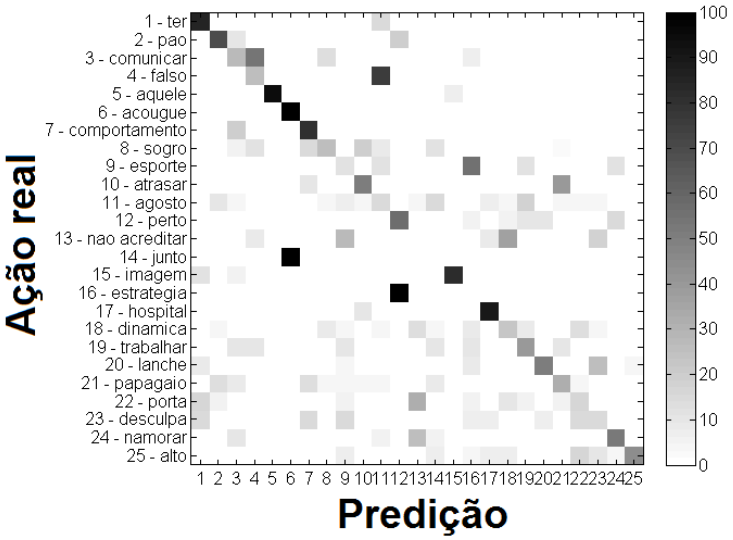
4.2.3 Ajuste de parâmetros: quantidade de pontos de interesse (m) e grupos de palavras visuais (k)

Para efeito de ajuste de parâmetros, adaptamos a implementação de [Laptev, 2004] para detecção de pontos de interesse e extração de características, alterando o número de pontos de interesse detectados em cada vídeo e verificando como isso interfere no resultado de classificação. A implementação original seleciona os pontos de interesse, a partir de um certo limiar, com maiores variações espaciais e temporais.

Alguns valores de m foram testados e os resultados mais interessantes foram encontrados para 50 pontos de interesse. Isto é, foram selecionados os 50 pontos de interesse mais relevantes, com maior variação espacial e temporal, para serem armazenados



a) Matriz de confusão para grupo 1



b) Matriz de confusão para grupo 2

Figura 4.5. Matrizes de confusão para grupos de sinais. Em (a) o grupo 1, com sinais distintos. Em (b), o grupo 2 com sinais similares. A acurácia é exibida em escala cinza na diagonal, com branco (0%) até preto (100%). Células de cor cinza fora da diagonal indicam erros de classificação.

para cada vídeo. Além disso, o algoritmo utilizado no teste anterior foi novamente utilizado para diferentes quantidades de palavras visuais e os melhores resultados foram encontrados para $k = 50$. Isso foi feito utilizando os mesmos 10 sinais do teste anterior e o classificador LDA, que foi selecionado com base nos seus resultados. As várias

quantidades de pontos de interesse (m) e grupos de palavras visuais (k) foram testados e os resultados são apresentados na Tabela 4.4, incluindo a configuração original de [Laptev, 2004].

Tabela 4.4. Acurácia do classificador LDA para bases de dados com diferentes quantidades de pontos de interesse (m) e grupos (k).

m	$k = 10$	$k = 20$	$k = 30$	$k = 50$	$k = 70$	$k = 100$
Original	42.00	55.13	62.13	65.26	57.93	51.53
20	40.46	53.80	62.66	67.46	62.86	53.93
30	46.26	56.00	63.86	70.00	65.86	54.13
50	53.53	63.73	67.53	**72.00**	64.20	54.80
70	50.60	63.26	66.06	67.20	60.60	47.80
100	50.73	61.13	64.40	63.86	61.53	48.40

Percebemos que ocorre um aumento de aproximadamente 7% na acurácia do classificador utilizando a base de dados adaptada com $m = 50$ em relação à implementação original de [Laptev, 2004]. No entanto, se aumentarmos para $m = 70$ e valores superiores, a acurácia diminui. Sendo assim, o parâmetro $m = 50$ foi definido para os próximos experimentos. Da mesma forma, em relação ao número de palavras visuais (k), há um ganho até $k = 50$. A partir desse valor, o desempenho do classificador não tem ganho e passa a ter perda. Então, fixamos os parâmetros $m = 50$ e $k = 50$ para os experimentos seguintes.

4.2.4 Conjunto de descritores de vídeos com parâmetros ajustados

Para anular a aleatoriedade gerada na tarefa de agrupamento, diminuir o tempo de execução do algoritmo e criar um ambiente mais interessante para realização dos experimentos seguintes, foi criado um conjunto de descritores dos vídeos com os parâmetros ajustados ($m = 50$ e $k = 50$). Esse conjunto foi gerado a partir de 100 execuções do *K-means*, utilizando os 1500 arquivos de descritores espaço-temporais. O agrupamento que gerou o melhor resultado de classificação foi escolhido e os histogramas, gerados a partir da contagem de palavras visuais resultante desse processo, foram armazenados em uma base de descritores dos vídeos.

4.3 2º Configuração experimental

Trabalhos de classificação com abordagens bioinspiradas têm atenção crescente em diversos campos [Xue et al., 2016; Huerta et al., 2010; Ribeiro et al., 2015], mas

ainda não foram muito exploradas no contexto do reconhecimento de gestos. Como esse trabalho trata especificamente do reconhecimento de sinais da Libras, tarefa que tem se apresentado desafiadora para os classificadores convencionais, optamos por realizar um experimento utilizando um classificador de abordagem baseada na combinação do ClonALG [de Castro & Zuben, 2002] com o algoritmo de treinamento da rede neural de Kohonen [Kohonen, 2001]. Pretendemos, então, comparar o desempenho entre a LDA, classificador que obteve melhor resultado em comparação com outros da biblioteca *Scikit Learn*, e o classificador Imune/neural [D'Angelo et al., 2016].

Nesse experimento, assim como no anterior, para cada tipo de sinal foi selecionado um vídeo para teste e os outros 14 para treinamento. O experimento foi executado 15 vezes, de modo que em cada vez um vídeo diferente foi selecionado para o teste e o restante para o treinamento, em uma estratégia de *15-fold*. Como resultado da taxa de reconhecimento do classificador, foi calculada a média das 15 execuções e desvio padrão para cada sinal. Foi utilizado o conjunto de descritores dos vídeos com os parâmetros ajustados ($m = 50$ e $k = 50$).

4.3.1 classificador Imune/neural x LDA

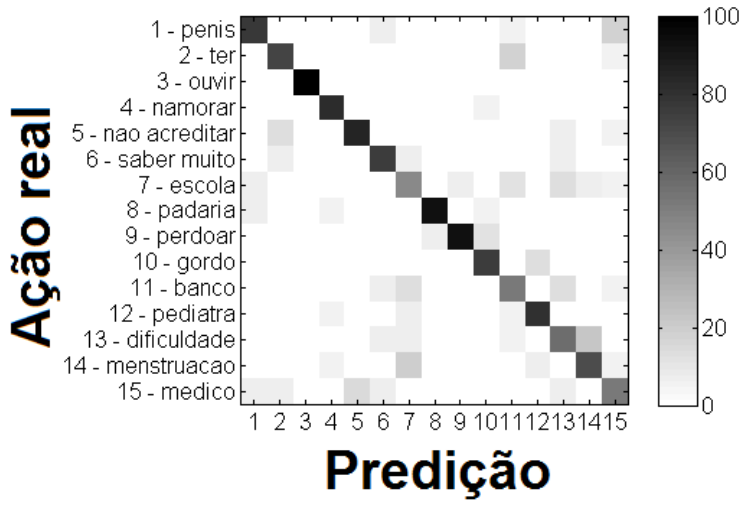
No primeiro teste desse experimento, foram selecionados 15 sinais para classificação, considerados de baixa ambiguidade. Em seguida, foram apresentados a acurácia e o *precision/recall* para cada sinal. Os sinais selecionados e a respectiva acurácia para LDA e o método Imune/neural, sob a forma de matrizes de confusão, são apresentados na Figura 4.6.

A acurácia total para LDA foi de 75.11%, enquanto que o Imune/neural obteve 82.22%. A partir de matrizes de confusão, foram calculados os valores de *precision* e *recall* para cada sinal. Cada valor na diagonal de uma matriz de confusão é chamado verdadeiro positivo (TP). Os valores fora da diagonal ao longo de uma coluna são chamados de falsos positivos (FP) e valores fora da diagonal ao longo de uma linha são chamados falsos negativos (FN). Enquanto que a *precision* mede a capacidade do classificador para evitar FP, o *recall* é a capacidade de evitar FN. Formalmente, em termos de TP, FP e FN, temos, para uma matriz de confusão C, que

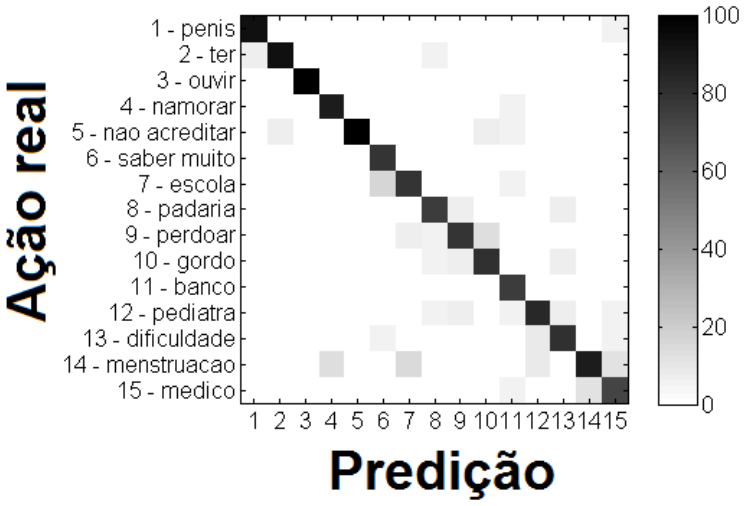
$$Precision = \frac{TP}{TP + FP} = \frac{C_{ii}}{\sum_i C_{ij}} \tag{4.1}$$

$$Recall = \frac{TP}{TP + FN} = \frac{C_{ii}}{\sum_j C_{ij}}. \tag{4.2}$$

Na Tabela 4.5, valores de *precision* e *recall* são apresentados para cada sinal



a) Matriz de confusão para LDA



b) Matriz de confusão para Imune/neural

Figura 4.6. Matrizes de confusão para: (a) LDA e (b) Imune/neural.

individual, tanto para o LDA quanto para o Imune/neural. Os melhores resultados foram destacados em negrito. Verificamos que, em quase todos os casos, o Imune/neural obteve os melhores valores de *precision* e *recall*.

Sinal	LDA		Immune / Neural	
	Precision	Recall	Precision	Recall
1-pênis	76.92	72.09	**93.33**	**94.38**
2-ter	73.33	75.71	**92.86**	**88.09**
3-ouvir	100.00	100.00	**100.00**	**100.00**
4-namorar	82.35	93.33	**87.50**	**94.59**
5-não acreditar	84.62	76.87	**100.00**	**84.17**
6-saber muito	75.00	79.30	**78.95**	**100.00**
7-escola	46.67	46.72	**78.57**	**79.08**
8-padaria	**92.31**	82.59	76.47	**84.70**
9-perdoar	**92.31**	**82.59**	78.57	74.88
10-gordo	76.47	**85.15**	**80.00**	80.25
11-banco	52.94	58.24	**75.00**	**100.00**
12-pediatra	80.00	**81.27**	**83.33**	73.37
13-difícil	56.25	57.32	**80.00**	**80.68**
14-menstruação	69.23	64.30	**88.89**	**65.79**
15-médico	52.94	55.62	**72.22**	**81.76**

Tabela 4.5. Comparação do *precision/recall*, em (%), para cada sinal usando LDA e Imune/neural. Os melhores resultados estão destacados em negrito.

4.3.2 Escalabilidade

No segundo teste desse experimento, foi investigada a escalabilidade dos classificadores para um número cada vez maior de sinais. Considerando apenas dois sinais, tanto o LDA como o Imune/neural obtiveram 100% de acerto de classificação. Ao serem considerados três sinais diferentes, o Imune/neural ainda foi capaz de obter 100% de acerto, enquanto que usando LDA a acurácia diminuiu para 95,55%. Com o aumento do número de sinais, observamos uma diminuição da acurácia tanto para LDA quanto para o Imune/neural. No entanto, como podemos ver na Figura 4.7, a acurácia obtida com o Imune/neural foi sempre maior do que com LDA. Observamos que, com maior número de sinais, a diferença de acertos foi próxima de 10%. Por exemplo, quando consideramos 25 sinais, o Imune/neural obteve 66,93%, enquanto LDA foi apenas 57,06%.

Finalmente, foram considerados os 100 sinais para treinamento e testes. A acurácia média de reconhecimento global do Imune/neural foi de 40,66% e utilizando o LDA a acurácia foi de 25,60%. Nesse nível, considerado o mais difícil para a base em questão, já foi possível verificar que o classificador Imune/neural apresentou desempenho consideravelmente melhor do que o LDA.

Os testes realizados demonstraram a superioridade do classificador Imune/neural, em comparação com o LDA. O uso de algoritmos evolutivos e Visão Computacional

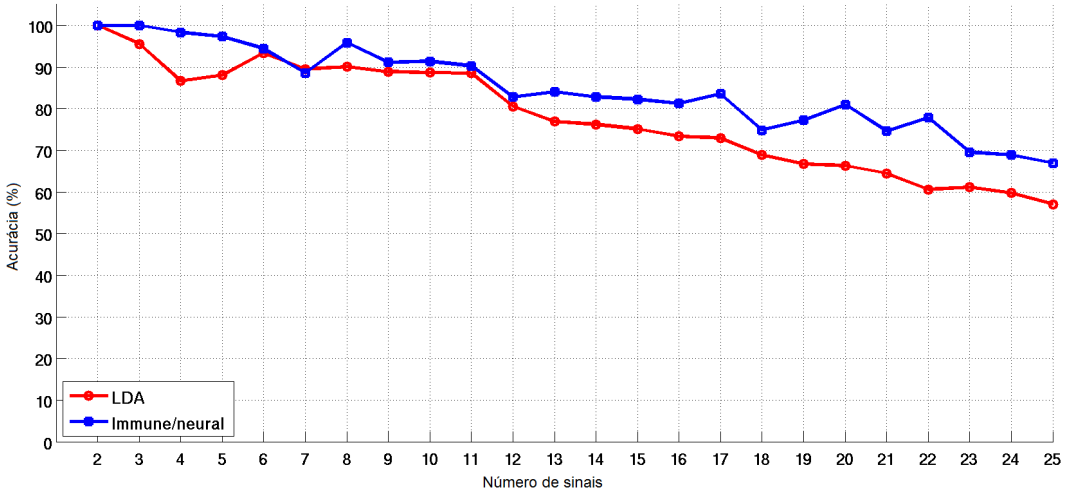


Figura 4.7. Gráfico de evolução da acurácia com o aumento do número de sinais.

mostrou ser uma estratégia promissora quando aplicada ao reconhecimento da língua de sinais a partir de sequências de vídeo.

Capítulo 5

Conclusão

O reconhecimento de línguas de sinais em vídeo, atualmente, ainda é um desafio. Nossa revisão bibliográfica mostra que existe um esforço da comunidade científica no sentido de automatizar a tradução da língua de sinais, tanto no Brasil, com relação à Libras, como nos demais países com suas respectivas línguas de sinais. Entretanto, os resultados ainda são tímidos e se justificam todas as iniciativas no sentido de contribuir na direção do desenvolvimento de uma solução completa que traria inúmeros benefícios para inclusão social de pessoas surdas.

Nosso trabalho buscou contribuir com o desenvolvimento de um descritor para sinais capturados em vídeo de forma a dispensar dispositivos mais elaborados como *scanner* 3D ou luvas com sensores de movimento, de forma que a solução se torna mais viável para ser embarcada em dispositivos móveis que são de fácil acesso à comunidade em geral. Isso foi possível com a utilização de um descritor que captura, em vídeo, as dinâmicas espacial e temporal dos movimentos.

Além disso, diferentemente da maioria dos trabalhos que são validados usando sinais estáticos relacionados a postura fixa das mãos para indicar números, letras do alfabeto e sinais simples, nossa abordagem foi validada usando sinais dinâmicos e complexos que envolvem, além da dinâmica dos movimentos das mãos, a postura corporal e expressões não manuais. Essa característica torna a abordagem mais abrangente em relação à diversidade de sinais da Libras. Para isso, construímos uma base de dados de vídeo com 100 sinais diferentes da Libras, executados por pessoas fluentes em Libras, num total de 1500 amostras de vídeos para experimentos de validação da nossa abordagem. Essa base de dados poderá ser usada em trabalhos futuros e por outros pesquisadores para comparação de resultados e melhoria dos métodos, pois verificamos que a maioria dos trabalhos da literatura considera conjuntos limitados de sinais e, na sua maioria, executados por pessoas não fluentes em Libras.

Finalmente, durante os experimentos, exploramos diversos métodos de classificação da literatura e verificamos que uma formulação híbrida Imune/neural pode ser usada para aumentar o desempenho de classificação em comparação com métodos de separação linear como LDA. Os experimentos mostram que a performance dos classificadores diminuiu com o aumento do número de classes de sinais considerados, entretanto consegue acurácia de 100% com 2 ou 3 sinais e acurácia média superior a 90% para testes com até 10 sinais distintos.

5.1 Limitações e trabalhos futuros

Os objetivos estabelecidos para o presente trabalho foram alcançados sem que se esgote o tema abordado. Resultados importantes foram obtidos, mas o desafio continua, especialmente em relação às limitações da abordagem. Nesse sentido, enumeramos algumas limitações e direções futuras para continuidade do trabalho:

- O algoritmo utilizado para detecção dos pontos de interesse é baseado no detector Harris-Laplace, conforme proposto por [Laptev, 2005]. Outros detectores podem ser explorados, investigando o impacto desses no desempenho dos classificadores.

- Os classificadores da biblioteca *Scikit-Learn* foram usados com configuração padrão. Como os vários classificadores funcionam de forma diferente e possuem diferentes parâmetros de entrada, alguns parâmetros podem ser ajustados e seus resultados confrontados com os resultados atuais.

- A acurácia dos classificadores é alta para pequeno número de classes de sinais, mas diminuiu à medida que se aumenta a quantidades de sinais. Testes com outros descritores ou ajustes na implementação do STIP, além de ajuste de parâmetros objetivando um melhor desempenho para grandes quantidades de sinais, podem ser realizados.

- O desempenho dos classificadores diminuiu também para grupos de sinais similares com alguma ambiguidade. Um trabalho com uma combinação de classificadores pode ser realizado, já que diferentes classificadores têm desempenho distinto para o mesmo grupo de sinais.

- Apesar dos bons resultados do classificador Imune/neural, o custo computacional dessa abordagem limitou a quantidade de experimentos. Uma possibilidade de trabalho futuro é utilizar a abordagem Imune/neural com maior tempo de execução, usando equipamentos de maior capacidade, para aumentar o número de

iterações definido como critério de parada, para verificar se ocorre aumento na acurácia.

- Devido a grande quantidade de sinais a serem treinados para uma aplicação real, uma árvore de prefixos pode ser implementada para diminuir a carga de treinamento, como em [Kadir et al., 2004].

- Não é possível utilizar a implementação em ambientes de tempo real. Adaptações na implementação, com otimização, e melhores resultados no desempenho do sistema são necessários para o desenvolvimento de uma aplicação tradutora de Libras em tempo real.

Referências Bibliográficas

Almeida, S. G. M.; Guimaraes, F. G. & Ramirez, J. A. (2014). Feature extraction in brazilian sign language recognition based on phonological structure and using RGB-D sensors. *Expert Systems with Applications*, 41(16):7259 -- 7271.

Anetha, K. & Parvin, J. R. (2014). Hand talk a sign language recognition based on accelerometer and SEMG data. *International Journal of Innovative Research in Computer and Communication Engineering*, 2(Special Issue NSCI 2014):206--215.

Bastos, I. L. O.; Angelo, M. F. & Loula, A. C. (2015). Recognition of static gestures applied to brazilian sign language (libras). Em *2015 28th SIBGRAPI Conference on Graphics, Patterns and Images*, pp. 305–312.

Bay, H.; Ess, A.; Tuytelaars, T. & Gool, L. V. (2008). Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110:346--359.

Bowden, R.; Windridge, D.; Kadir, T.; Zisserman, A. & Brady, M. (2004). A linguistic feature vector for the visual interpretation of sign language. Em *Proceedings of the 8th European Conference on Computer Vision (ECCV'04)*, pp. 390--401.

Bragatto, T. A. C.; Ruas, G. I. S.; Martins, A. P. & Lamar, M. V. (2009). Um modelo de máquinas de vetores de suporte estruturadas em árvore binária probabilística aplicado ao reconhecimento de posturas manuais em tempo real. Em *XXVII SIMPÓSIO BRASILEIRO DE TELECOMUNICACÕES - SBrT*.

Brasil (2002). Lei 10.436, de 24 de abril de 2002. dispõe sobre a língua brasileira de sinais - libras e dá outras providências. [Online; acessado em Junho-2017].

Brasil (2005). Decreto 5.626, de 22 de dezembro de 2005. regulamenta a lei no 10.436, de 24 de abril de 2002, que dispõe sobre a língua brasileira de sinais - libras, e o art. 18 da lei no 10.098, de 19 de dezembro de 2000. [Online; acessado em Junho-2017].

Capovilla, F. C.; Raphael, W. D. & Mauricio, A. C. L. (2013). *NOVO DEIT-LIBRAS: Dicionário Enciclopédico Ilustrado Trilíngue da Língua de Sinais Brasileira (libras) Baseado em Linguística e Neurociências Cognitivas*. EDUSP, 3st edição.

Cardenas, E. E. & Chavez, G. C. (2015). A robust gesture recognition using hand local data and skeleton trajectory. Em *2015 IEEE International Conference on Image Processing (ICIP)*, pp. 1240–1244.

Caridakis, G.; Diamanti, O.; Karpouzis, K. & Maragos, P. (2008). Automatic sign language recognition: Vision based feature extraction and probabilistic recognition scheme from multiple cues. Em *Proceedings of the 1st International Conference on PErvasive Technologies Related to Assistive Environments*, pp. 89:1--89:8.

Carneiro, A. T. S.; Cortez, P. C. & Costa, R. C. S. (2009). Reconhecimento de gestos da libras com classificadores neurais a partir dos momentos invariantes de hu. Em *Interaction South America 09*.

Carneiro, S. B.; Santos, E. D. F. M.; Barbosa, T. M. A.; Ferreira, J. O.; AlcalÃ¡, S. G. S. & da Rocha, A. F. (2016). Static gestures recognition for brazilian sign language with kinect sensor. Em *2016 IEEE SENSORS*, pp. 1–3.

Chen, F.-S.; Fu, C.-M. & Huang, C.-L. (2003). Hand gesture recognition using a real-time tracking method and hidden markov models. *Image and Vision Computing*, 21(8):745 – 758.

Cooper, H.; Ong, E. J.; Pugeault, N. & Bowden, R. (2012). Sign language recognition using sub-units. *Journal of Machine Learning Research*, 13:2205--2231.

da Silva Júnior, J. P. (2014). Alinhamento de imagens de profundidade com aplicação no reconhecimento da língua de sinais. Dissertação de mestrado, Universidade de Brasília.

D'Angelo, M. F. S. V.; Palhares, R. M.; Filho, M. C. O. C.; Maia, R. D.; Mendes, J. B. & Ekel, P. Y. (2016). A new fault classification approach applied to tennessee eastman benchmark process. *Applied Soft Computing*, 49:676--686.

Dawn, D. D. & Shaikh, S. H. (2016). A comprehensive survey of human action recognition with spatio-temporal interest point (STIP) detector. *The Visual Computer: International Journal of Computer Graphics*, 32(3):289--306.

de Castro, L. & Zuben, F. (2002). Learning and optimization using the clonal selection principle. *IEEE Transactions on Evolutionary Computation*, 6(3):239--251.

de Paula Neto, F. M.; Cambuim, L. F.; Macieira, R. M.; Ludermir, T. B.; Zanchettin, C. & Barros, E. N. (2015). Extreme learning machine for real time recognition of brazilian sign language. Em *2015 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 1464–1469.

de Souza, C. R. & Pizzolato, E. B. (2013). Sign language recognition with support vector machines and hidden conditional random fields: Going from fingerspelling to natural articulated words. Em Perner, P., editor, *Machine Learning and Data Mining in Pattern Recognition*, volume 7988 of *Lecture Notes in Computer Science*, pp. 84--98. Springer.

de Souza, R. (2014). Reconhecimento de gestos usando distância de cadeias de descritores. Dissertação de mestrado, Pontifêcia Universidade Católica do Paraná.

Dong, C.; Leu, M. C. & Yin, Z. (2015). American sign language alphabet recognition using microsoft kinect. Em *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 44–52.

dos Santos, J. R. (2015). Reconhecimento das configurações de mão de libras baseado na análise de discriminante de fisher bidimensional utilizando imagens de profundidade. Dissertação de mestrado, Universidade Federal do Amazonas, Manaus, AM.

Harris, C. & Stephens, M. (1988). A combined corner and edge detector. Em *In Proc. of Fourth Alvey Vision Conference*, pp. 147--151.

Huerta, E. B.; Duval, B. & Hao, J. (2010). A hybrid lda and genetic algorithm for gene selection and classification of microarray data. *Neurocomputing*, 73(13–15):2375--2383.

Idrees, H.; Zamir, A. R.; Jiang, Y.; Gorban, A.; Laptev, I.; Sukthankar, R. & Shah, M. (2017). The THUMOS challenge on action recognition for videos in the wild. *Computer Vision and Image Understanding*, 155:1--23.

Iosifidis, A.; Tefas, A. & Pitas, I. (2015). Merging linear discriminant analysis with bag of words model for human action recognition. Em *2015 IEEE International Conference on Image Processing (ICIP)*, pp. 832–836.

Jangyodsuk, P.; Conly, C. & Athitsos, V. (2014). Sign language recognition using dynamic time warping and hand shape distance based on histogram of oriented gradient features. Em *Proceedings of the 7th International Conference on PErvasive Technologies Related to Assistive Environments*, p. 50.

Kadir, T.; Bowden, R.; OngJ., E. & Zisserman, A. (2004). Minimal training, large lexicon, unconstrained sign language recognition. Em *British Machine Vision Conference*, pp. 1--10.

Koenderink, J. J. & van Doorn, A. J. (1992). Generic neighborhood operators. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 597--605.

Kohonen, T. (2001). *Self-organizing maps*. Springer.

Kuroda, T.; Tabata, Y.; Goto, A.; Ikuta, H. & Murakami, M. (2004). Consumer price data glove for sign language recognition. Em *Proceedings of the 5th International Conference on Disability, Virtual Reality and Assoc. Tech.*, pp. 253–258.

Laptev, I. (2004). Local spatio-temporal image features for motion interpretation. Dissertação de mestrado, KTH Numerical Analysis and Computer Cience, Stockholm.

Laptev, I. (2005). On space-time interest points. *IJCV - International Journal of Computer Vision*, 64(2–3):107--123.

Liu, M.; Liu, H.; Sun, Q.; Zhang, T. & Ding, R. (2016). Salient pairwise spatio-temporal interest points for real-time activity recognition. *CAAI - Transactions on Intelligence Technology*, 1(1):14--29.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91--110.

Marcotti, P.; Abiuzi, L. B.; Rizol, P. M. S. R. & Forster, C. H. Q. (2007). Interface para reconhecimento da língua brasileira de sinais. Em *XVIII Simpósio de Informática na Educação - SBIE Workshop em Brasileiro Informática na Educação.*

Neto, J. P. S. & Oquendo, L. (2013). Estudo do estado da arte das técnicas de reconhecimento das línguas de sinais por computador. Em *Proeceedings of the InfoBrasil 2013.*

Patel, C. I.; Garg, S.; Zaveri, T.; Banerjee, A. & Patel, R. (2016). Human action recognition using fusion of features for unconstrained video sequences. *Computers e Electrical Engineering.*

Pavan, A. R. & Modesto, F. A. C. (2010). Reconhecimento de gestos com segmentação de imagens dinâmicas aplicadas a libras. *Biblioteca Digital Brasileira de Computação.*

Ribeiro, L. A.; Soares, A. S.; Lima, T. W.; Jorge, C. A. C.; da Costa, R. M.; Salvini, R. L.; Coelho, C. J.; Federson, F. M. & Gabriel, P. H. R. (2015). Multi-objective genetic algorithm for variable selection in multivariate classification problems: A case study in verification of biodiesel adulteration. *Procedia Computer Science*, 51:346--355.

Siola, F. B. (2010). Desenvolvimento de um software para tradução de libras/português. Em *III SIMPÓSIO DE INICIACÃO CIENTÍFICA*.

Souza, K. P.; Dias, J. B. & Pistori, H. (2007). Reconhecimento automático de gestos da língua brasileira de sinais utilizando visão computacional. Em *III WVC - Workshop de Visão Computacional*.

Souza, M. C. T.; Teixeira, E. M.; Winagraski, E. & Castro, H. C. (2016). Sign language, written language, hearing parents and books: Together for the good of deaf children cognitive and emotional development. *Arts, Literature and Linguistics*, 2(1):1--10.

Subetha, T. & Chitrakala, S. (2016). A survey on human activity recognition from videos. Em *2016 International Conference on Information Communication and Embedded Systems (ICICES)*, pp. 1–7.

Wanderlan, C.; Tenório, R. & Luz, T. (2013). Hand talk - mobile app. [Online; acessado em Junho-2017].

Wang, H.; Chai, X. & Chen, X. (2016). Sparse observation (SO) alignment for sign language recognition. *Neurocomputing*, 175, Part A:674--685.

Willems, G.; Tuytelaars, T. & Gool, L. (2008). An efficient dense and scale-invariant spatio-temporal interest point detector. Em *Proceedings of the 10th European Conference on Computer Vision: Part II*.

Xue, B.; Zhang, M.; Browne, M. W. N. & Yao, X. (2016). A survey on evolutionary computation approaches to feature selection. *IEEE Transactions on Evolutionary Computation*, 20(4):606--626.

Yan, Y.; Ricci, E.; Subramanian, R.; Liu, G. & Sebe, N. (2014). Multitask linear discriminant analysis for view invariant action recognition. *IEEE Transactions on Image Processing*, 23(12):5599--5611.

Yang, J.; Jiang, Y.-G.; Hauptmann, A. G. & Ngo, C.-W. (2007). Evaluating bag-of-visual-words representations in scene classification. Em *Proceedings of the International Workshop on Workshop on Multimedia Information Retrieval*.

Yang, Q. (2010). Chinese sign language recognition based on video sequence appearance modeling. Em *Proceedings of the 5th IEEE Conference on Industrial Electronics and Applications*, pp. 1537–1542.

Zhao, Y.; Di, H.; Zhang, J.; Lu, Y.; Lv, F. & Li, Y. (2017). Region-based mixture models for human action recognition in low-resolution videos. *Neurocomputing*, pp. 1--15.

Anexo A

Análise discriminante linear - LDA

[1] Ambos LDA e QDA podem ser derivados de modelos probabilísticos simples que modelam a distribuição condicional de classes dos dados $P(X|y = k)$ para cada classe k. Predições podem então ser obtidas usando a regra de Bayes:

$$P(y = k|X) = \frac{P(X|y = k)P(y = k)}{P(X)} = \frac{P(X|y = k)P(y = k)}{\sum_l P(X|y = l) \cdot P(y = l)} \tag{A.1}$$

Então, selecionamos a classe k que maximiza essa probabilidade condicional.

Mais especificamente, para análise de discriminantes linear e quadrático, $P(X|y)$ é modelado como uma distribuição gaussiana multivariada com densidade:

$$P(X|y = k) = \frac{1}{(2\pi)^n |\Sigma_k|^{1/2}} \exp\left(-\frac{1}{2}(X - \mu_k)^t \Sigma_k^{-1} (X - \mu_k)\right) \tag{A.2}$$

Para usar esse modelo como classificador, precisamos apenas estimar nos dados de treinamento a priore $P(y = k)$ (pela proporção de instâncias da classe k), as médias das classes μ_k (pela média da amostragem empírica da classe) e as matrizes de covariância (pela matriz de covariância da amostragem empírica da classe, ou por um estimador regularizado).

No caso do LDA, as gaussianas para cada classe são consideradas como compartilhando a mesma matriz de covariância $\Sigma_k = \Sigma$ para todos os k. Isto leva a superfícies de decisão lineares entre classes, como pode ser visto comparando as relações de log-probabilidade $\log[P(y = k|X)/P(y = l|X)]$.

$$\log\left(\frac{P(y = k|X)}{P(y = l|X)}\right) = 0 \Leftrightarrow (\mu_k - \mu_l)\Sigma^{-1} X = \frac{1}{2}(\mu_k^t \Sigma^{-1} \mu_k - \mu_l^t \Sigma^{-1} \mu_l) \tag{A.3}$$

No caso do QDA, não há essas considerações sobre as matrizes de covariância Σ_k das gaussianas, levando a superfícies quadráticas de decisão.

[1] reproduzido de http://scikit-learn.org/stable/modules/lda_qda.html