Geovane Menezes Ramos Neto

Reconhecimento de Língua de Sinais Baseado em Redes Neurais Convolucionais 3D

Universidade Federal do Maranhão

Centro de Ciências Exatas e Tecnológicas

Curso de Ciência da Computação

São Luís - MA

2018

Geovane Menezes Ramos Neto

Reconhecimento de Língua de Sinais Baseado em Redes Neurais Convolucionais 3D

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da UFMA como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciência da Computação.

Universidade Federal do Maranhão

Centro de Ciências Exatas e Tecnológicas

Curso de Ciência da Computação

Orientador: Prof. Dr. Geraldo Braz Junior

São Luís - MA

2018

Agradecimentos

A Deus, pois me proporciona saúde a cada manhã.

Aos meus pais, Elizabeth Araújo dos Passos Ramos e Giovanni Ramos Filho, pelo carinho, suporte e dedicação.

Ao meu irmão João Gabriel dos Passos Ramos, por seu companheirismo e estimulo a sempre estudar e me aprimorar mais.

A minha namorada Alessandra de Jesus Fernandes Carvalho, pelo seu afeto e dedicação, que está sempre disposta a ajudar e cuja opinião me abre a mente e mostra as perspectivas que sozinho eu nunca veria.

Ao meu orientador Geraldo Braz Junior, que me motiva através de seu exemplo, cujo incentivo me fez terminar os projetos que iniciei.

A minha amiga Maria Angélica Paixão Frazão, por seu carinho e por sempre acreditar em mim, que enxergou um potencial que ninguém mais viu.

Ao meu amigo Matheus Chaves Menezes que, além do companheirismo, me fez entender que a vida não é um filme e sim um jogo.

Aos professores do programa de pós-graduação em Ciência da Computação, por todo conhecimento transmitido em sala de aula.

E a todos que direta ou indiretamente fizeram parte da minha formação, serei eternamente grato.

"Estabilidade não existe.
A vida é um risco a partir do instante em que você chegou a este mundo."
(Flávio Augusto)

Resumo

A necessidade em utilizar um código linguístico prioritariamente visual dificulta o desenvolvimento de indivíduos com deficiência auditiva. Esta dificuldade é explicada pela baixa quantidade de pessoas fluentes em uma língua de sinais, limitando a inclusão dos deficientes auditivos. As soluções atuais para a comunicação entre pessoas sem o domínio de uma língua de sinais e deficientes auditivos são a utilização de tradutores humanos, que são recursos onerosos devido a experiência profissional necessária.

Este estudo apresenta uma metodologia que utiliza técnicas de visão computacional e aprendizado de máquina para reconhecer sinais da Língua de Sinais Argentina. O reconhecimento se dá através da utilização de uma arquitetura 3D *Convolutional Neural Network*, que foi construída através da seleção dos parâmetros que forneceram os melhores resultados entre os testes realizados. Para a validação, utilizamos a base de vídeos LSA64, que contem 64 sinais da Língua de Sinais Argentina. A melhor arquitetura alcançou uma acurácia média de $94,22\%$ que, quando comparado a trabalhos relacionados, se mostrou uma metodologia promissora no reconhecimento automático de línguas de sinais.

Palavras-chave: Reconhecimento de Língua de Sinais; Rede Neural Convolucional 3D; Aprendizado Profundo

Abstract

The need to use a visual language code makes the development of hearing impaired individuals difficult. This difficulty is explained by the low number of people who are fluent in a sign language, limiting the inclusion of the hearing impaired. The current solutions for communication between people without the domain of sign language and the hearing impaired are the use of human translators, which are expensive resources due to the necessary professional experience.

This study presents a methodology that uses computer vision and machine learning techniques to recognize signals from the Sign Language of Argentina. The recognition takes place through the use of a 3D Convolutional Neural Network architecture, which was built through the selection of the parameters that provided the best results among the tests performed. For validation, we use the LSA64 video base, which contains 64 signs of the Sign Language Argentina. The best architecture achieved an average accuracy of 94.22% which, when compared to related works, proved to be a promising methodology in the automatic recognition of sign languages.

Keywords: Sign Language Recognition; 3D *Convolutional Neural Network*; *Deep Learning*

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

ACM	*Association for Computing Machinery*
API	*Application Programming Interface*
BOW	*Bag of Words*
BHOF	*Block-based Histogram of Oriented Gradients*
BLSTMNN	*Bidirectional Long Short-Term Memory Neural Network*
CHMM	*Coupled Hidden Markov Model*
CRF	*Conditional Random Field*
CNN	*Convolutional Neural Network*
ELMNN	*Extreme Learning Machine Neural Network*
EMG	Eletromyograma
FC	*Fully Connected*
HMM	*Hidden Markov Model*
HOD	*Histogram of Oriented Displacements*
HOG	*Histogram of Oriented Gradients*
HSV	*Hue, Saturation, Value*
IA	Inteligência Artificial
IEEE	*Institute of Electrical and Electronic Engineers*
LS	Língua de Sinais
LSA	Língua de Sinais Argentina
LSTM	*Long Short-Term Memory*
NNI	*Nearest Neighbor Interpolation*
SIFT	*Scale Invariant Feature Transform*
SOM	*Self Organized Maps*

SVM	*Support Vector Machine*
PCA	*Principal Component Analysis*
ReLU	*Rectified Linear Unit*
RGB	*Red, Green, Blue*
RGBD	*Red, Green, Blue, Depth*
RNN	*Recurrent Neural Network*
RProp	*Resilient Propagation*
RMSProp	*Root Mean Square Propagation*

Sumário

1 Introdução

A inclusão de deficientes ainda é um problema recorrente em todo o mundo. No caso da deficiência auditiva o problema está na dificuldade que estes têm para comunicar-se utilizando um código de linguagem prioritariamente visual, uma Língua de Sinais (LS). Uma das grandes dificuldades para os deficientes auditivos é a baixa quantidade de pessoas que são fluentes em língua de sinais, o que dificulta o aprendizado bem como a comunicação destes, principalmente em estágios iniciais do desenvolvimento do indivíduo (MOELLER, 2000; DALTON et al., 2003).

Atualmente as soluções para o reconhecimento de uma LS se dão prioritariamente pela utilização de tradutores humanos e, portanto, são soluções caras dada a experiência profissional necessária. O reconhecimento de língua de sinais busca desenvolver algoritmos e métodos que possam reconhecer uma sequência de sinais para assimilar seu sentido. As LS são conjuntos estruturados de gestos onde cada gesto tem significado próprio além de regras de contexto sólidas, tais gestos são denominados sinais. A comunicação se dá através de variações das poses das mãos, movimentos corporais e até expressões faciais (STARNER; WEAVER; PENTLAND, 1998).

Podemos categorizar os sinais em dois tipos: estáticos e dinâmicos. Os sinais estáticos são representados por uma única pose das mãos e ausência de movimentos corporais e de mudanças nas expressões faciais durante a execução do sinal. Já os sinais dinâmicos tem a mudança das poses das mãos, movimentos corporais e expressões faciais como fatores que compõem o significado dos sinais (MITRA; ACHARYA, 2007).

Reconhecer sinais dinâmicos é uma tarefa complexa que envolve muitos aspectos como a modelagem de movimento, análise de movimento, reconhecimento de padrões, aprendizado de máquina e até mesmo estudos psicolinguísticos (WU; HUANG, 1999). Dada essa complexidade, muitas metodologias tratam o reconhecimento de sinais como um problema de reconhecimento de gestos, ou seja, as soluções buscam identificar características ótimas que representem satisfatoriamente um determinado gesto e métodos que possam classificá-lo corretamente dado um conjunto de sinais possíveis.

Neste trabalho pretendíamos utilizar uma base de vídeos da Língua Brasileira de Sinais (Libras), porém, devido a escassez de opções, não encontramos uma base pública disponível que se encaixasse nos propósitos deste trabalho e, por isso, optamos por utilizar uma base de vídeos da Língua de Sinais Argentina (LSA).

1.1 Objetivos

O objetivo deste trabalho é propor um método computacional eficiente para o reconhecimento de palavras pertencentes a línguas de sinais através de técnicas de visão computacional e aprendizado de máquina, no qual aplicaremos uma técnica de *deep learning* para o reconhecimento da sinais da LSA.

1.1.1 Objetivos Específicos

Precisamente busca-se:

- Implementar uma arquitetura 3D *Convolutional Neural Network* (3D CNN) (JI et al., 2013) e utiliza-la para representar e classificar 64 gestos dinâmicos da LSA presentes na base de vídeos LSA64 (RONCHETTI et al., 2016);
- Estudar as vantagens da utilização de uma arquitetura 3D para o reconhecimento de LS quando comparada aos métodos tradicionais e a outros tipos de arquitetura;

1.2 Contribuições

A principal contribuição deste trabalho é apresentar uma arquitetura 3D CNN ajustada através de treinamento para o reconhecimento da LSA, de forma que possa reconhecer seus gestos dinâmicos de maneira eficiente em comparação com outros trabalhos presentes na literatura. Neste trabalho focamos no reconhecimento de sinais dinâmicos dado que este pode ser facilmente estendido para sinais estáticos.

O trabalho apresenta ainda uma metodologia que pode ser utilizada para melhorar a qualidade de vida, inclusão e comunicação de deficientes auditivos facilitando a sua comunicação com pessoas que não tem o domínio sobre as línguas de sinais.

1.3 Organização da Dissertação

Esta dissertação está divida em 6 capítulos. No Capítulo 2 é apresentada a revisão sistemática bem como trabalhos relacionados adicionais. O Capítulo 3 apresenta a base de vídeos utilizada, as técnicas e os conceitos necessárias para o desenvolvimento deste trabalho.

O Capítulo 4 descreve todos os procedimentos realizados para reconhecer que sinais são apresentados em cada um dos vídeos da base utilizada. No Capítulo 5 estão dispostos os resultados obtidos através dos experimentos bem como a comparação destes com outros trabalhos presentes na literatura. Já no Capítulo 6 estão salientados os objetivos alcançados bem como os as limitações da metodologia proposta.

2 Trabalhos Relacionados

Entendendo a importância de conhecer algumas soluções propostas para o problema a que este trabalho se direciona, realizou-se uma revisão sistemática cujo principal objetivo é esclarecer quais as principais técnicas empregadas e que tipos de fatores são levados em consideração no desenvolvimento de um reconhecedor de linguagem de sinais. Neste estudo há duas maneiras de classificar os reconhecedores automáticos, sendo classificados pela natureza do gesto a ser reconhecido ou da forma com que obtemos os dados do gesto.

A revisão sistemática levou em consideração trabalhos disponíveis online nas bibliotecas digitais da ACM, IEEE Xplore e ScienceDirect da Elsevier, entre o período de 01 de Janeiro de 2000 até 26 de Março de 2017, escritos em qualquer estrutura, utilizando como termos de busca as palavras: *brazilian sign language recognition*, *sign language recognition* e *gesture recognition* utilizando o conectivo lógico de disjunção (OU) entre elas.

A seleção dos trabalhos foi feita em duas etapas. Na primeira etapa, os trabalhos foram analisados (através da leitura dos seus resumos) e excluídos quando se tratavam de publicações somente do resumo, revisões, trabalhos que não utilizam técnicas de visão computacional nem sensores, trabalhos cujo problema não seja reconhecer sinais ou trabalhos de apresentação de *frameworks*.

Os textos completos dos trabalhos que passaram pela primeira etapa ou dos trabalhos cujo resumo não foi suficiente para determinar sua natureza foram examinados na segunda etapa. A segunda etapa consiste na leitura dos textos completos e verificar se estes se enquadram na categorias de reconhecimento de gestos de línguas de sinais.

Dentre 60 artigos estudados, 49 passaram na primeira etapa pois se enquadraram nos critérios estabelecidos para a revisão. Enquanto que na segunda etapa, 33 artigos se categorizavam nos critérios de inclusão. Os trabalhos que satisfizeram os critério de inclusão da segunda etapa foram os selecionados para a revisão sistemática.

No decorrer do estudo, além da divisão em sinais estáticos e sinais dinâmicos, foi verificado que existe uma outra classificação e que esta se dá através de como os reconhecedores automáticos obtém informações sobre os gestos. Estas informações podem ser obtidas através de imagens de intensidade luminosa ou por meio de dados de sensores. A primeira delas utiliza as informações que podem ser extraídas de imagens cujos *pixels* representam informações visuais de uma cena (CHEN; GEORGANAS; PETRIU, 2007; HEAP; HOGG, 1996), enquanto a segunda utiliza informações como as provenientes de sensores como acelerômetros (XU; ZHOU; LI, 2012; LIU et al., 2009).

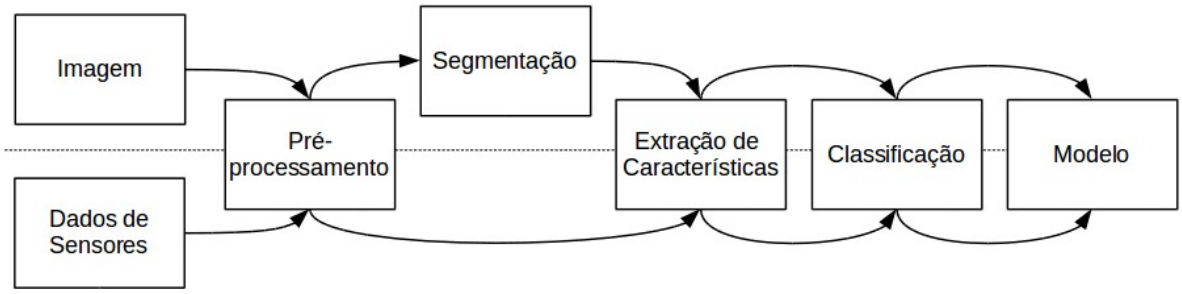
De forma genérica, as etapas dos trabalhos de reconhecimento de sinais estáticos

são apresentadas na Figura 1. Esta imagem é dividida por uma linha pontilhada, na qual etapas comuns a ambas estratégias estão sobre a linha pontilhada e etapas específicas estão apenas acima ou abaixo, dependendo da estratégia a que esta etapa pertence.

Praticamente todas as etapas são comuns a ambas estratégias, a não ser pela segmentação, que é utilizada apenas no reconhecimento de gestos estáticos utilizando imagens de intensidade luminosa. Este fluxo se inicia com uma imagem de um gesto a ser pré-processada, após o pré-processamento o gesto é então segmentado e tem suas características extraídas. Um classificador é treinado utilizando como entrada as características dos vários gestos de treino, criando um modelo para teste.

Por sua vez, as etapas do processo que utilizam as informações obtidas através de dados dos sensores, se iniciam com o pré-processamento e tem suas características extraídas a partir dos dados coletados. Essas características de vários sensores representam um gesto, o conjunto dessas representações é utilizado em um classificador para criar um modelo de teste.

Figura 1 – Modelos abstratos de etapas para o reconhecimento de gestos estáticos utilizando imagens de intensidade luminosa e dados de sensores.



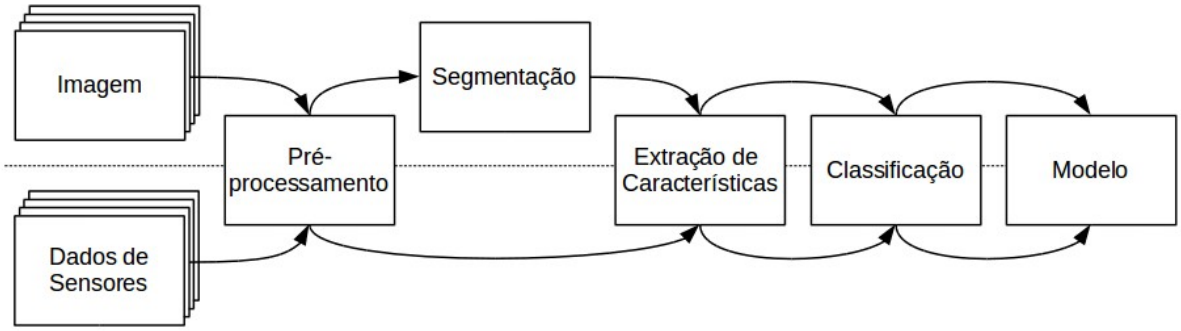
Fonte: Elaborada pelo autor.

Da mesma forma, os reconhecedores de gestos dinâmicos também tem seus modelos abstratos apresentados na Figura 2, dividindo os dois modelos da mesma maneira. Na parte superior, são apresentadas as mesmas etapas existentes para os gestos estáticos, porém um gesto será descrito por uma sequência temporal de imagens. Para toda imagem são extraídas características que, ao final, são integradas de tal maneira que representem o gesto e, só então, são utilizadas pelo classificador para criar o modelo para teste.

As etapas que utilizam dados de sensores dinâmicos, diferentemente das etapas para gestos estáticos, utilizam um conjunto de dados dos sensores que são capturados durante um intervalo de tempo para compor um conjunto de características que possam representar o gesto em capturado e, só então, são utilizados pelo classificador para criar o modelo de teste.

A seguir apresentaremos os trabalhos da revisão sistemática divididos em dois

Figura 2 – Modelos abstratos de etapas para o reconhecimento de gestos dinâmicos utilizando imagens de intensidade luminosa e dados de sensores.



Fonte: Elaborada pelo autor.

grupos, os que se propõem a reconhecer sinais estáticos e os que se propõe a reconhecer sinais dinâmicos. Dentro de cada grupo há uma subdivisão de trabalhos: trabalhos que utilizam imagens de intensidade luminosa e trabalhos que utilizam dados de sensores.

2.1 Sinais Estáticos

Dentre os trabalhos da revisão sistemática, 11 buscam reconhecer sinais estáticos. Dentre os trabalhos de sinais estáticos, 8 utilizam imagens de intensidade luminosa (NEIVA; ZANCHETTIN, 2016; WANG; CHEN; LI, 2016; NETO et al., 2015; BASTOS; ANGELO; LOULA, 2015; PORFIRIO et al., 2013; LIMA et al., 2012; BEDREGAL; DIMURO et al., 2006; DIAS et al., 2004). Dentre estes, o trabalho de (NETO et al., 2015) tem a proposta de minimizar o treinamento utilizando *Extreme Learning Machine Neural Network* (ELMNN) como classificador, limitando o uso de memória e processador além de aumentar a acurácia, utilizou como extrator de características o *Motion Grandient Orientation Image* e o *Principal Component Analysis* (PCA) para classificar 18 sinais estáticos da Libras.

Utilizando uma segmentação que utiliza 3 esquemas de cores (RGB, HSV, YIQ) e extraindo características com as técnicas *Histogram of Oriented Gradients* (HOG) e Momentos Invariantes de Zernique, o trabalho de (BASTOS; ANGELO; LOULA, 2015) utilizou uma divisão manual de grupos para realizar uma classificação em dois estágios utilizando *Multi-Layer Perceptron* (MLP) como classificador para reconhecer 40 sinais estáticos.

Utilizando um método para juntar projeções das mãos a fim de criar uma estrutura 3D, o método apresentado em (PORFIRIO et al., 2013) simplifica a estrutura e aplica uma transformada harmônica para extrair as características. Em seguida, classifica as 61 posições estáticas utilizando o classificador *Support Vector Machine* (SVM).

Por outro lado, os trabalhos que utilizam dados de sensores somam 4 (ABREU

et al., 2016; THALANGE; DIXIT, 2016; CARNEIRO et al., 2016; ANJO; PIZZOLATO; FEUERSTACK, 2012). A fim de reconhecer 20 sinais estáticos de Libras através dos sinais de Eletromyograma (EMG) da Myo *armband*, o trabalho de (ABREU et al., 2016) reduziu os efeitos do posicionamento dos sensores presentes na Myo *armband*, através de técnicas de pré-processamento e conseguiu resultados satisfatórios com o classificador SVM.

Carneiro et al. (2016) normaliza a iluminação de imagens de profundidade obtidas pelo sensor Kinect utilizando um limiar global, além de utilizar a técnica *eigenhands* (baseada na eigenfaces) buscando reconhecer 20 sinais estáticos da Libras.

Se tratando de trabalhos que visam reconhecer gestos estáticos, o único trabalho da revisão sistemática que utilizou uma combinação de imagens de intensidade luminosa e dados de sensores foi o trabalho de (CHANSRI; SRINONCHAT, 2016), que propõe um método robusto a mudanças de iluminação e fundo complexo utilizando imagens convencionais e de Kinect. A segmentação é baseada na profundidade dos objetos em cena e a técnica HOG é utilizada para extrair características de 66 classes de gestos estáticos do alfabeto tailandês.

Para fins de resumo, a Tabela 1 apresenta todos os trabalhos de reconhecimento de sinais estático utilizados nessa revisão sistemática, em termos de base e acurácia obtidos.

Tabela 1 – Tabela de resumo dos trabalho de reconhecimento de sinais estáticos, ordenados por ano de publicação e acurácia.

Trabalho	Acurácia (%)	#Classes	#Indivíduos
(ABREU et al., 2016)	99	20 letras	570.000 leituras
(WANG; CHEN; LI, 2016)	99	10 dígitos	30 imagens
(THALANGE; DIXIT, 2016)	98,17	10 dígitos	676 imagens
(CARNEIRO et al., 2016)	89	21 letras	480 imagens
(CHANSRI; SRINONCHAT, 2016)	84,05	24 sinais	720 imagens
(NEIVA; ZANCHETTIN, 2016)	84	12 letras	480 imagens
(BASTOS; ANGELO; LOULA, 2015)	96,77	40 sinais	9600 imagens
(NETO et al., 2015)	95,92	18 letras	990 imagens
(PORFIRIO et al., 2013)	96,83	61 posições	1220 imagens
(ANJO; PIZZOLATO; FEUERSTACK, 2012)	100	10 letras	400 imagens
(LIMA et al., 2012)	-	36 caracteres	-
(BEDREGAL; DIMURO et al., 2006)	-	-	-
(DIAS et al., 2004)	60	36 caracteres	-

2.2 Sinais Dinâmicos

Dos 33 trabalhos da revisão sistemática, 20 trabalhos tratam do reconhecimento de gestos dinâmicos. Os trabalhos que utilizam imagens de intensidade luminosa somam 7 (BELGACEM; CHATELAIN; PAQUET, 2017; LI; ZHOU; LEE, 2016; LIM; TAN; TAN, 2016b; MADEO et al., 2012; MADEO et al., 2010; DIAS et al., 2009; BRAGATTO; RUAS; LAMAR, 2006). O trabalho de (LIM; TAN; TAN, 2016b), por exemplo, utiliza a técnica

Block-based Histogram of Optical Flow (BHOF). Primeiramente, a face é descartada e utiliza filtros de média e mediana para detectar as mãos e construir o modelo de fundo. Baseado no modelo do fundo, a posição da mão é determinada e, posteriormente, as regiões das mãos são extraídas para então o histograma de fluxo óptico baseado em blocos ser computado para reconhecer 50 gestos dinâmicos.

Belgacem, Chatelain e Paquet (2017) propuseram um sistema Markoviano híbrido de *Conditional Random Field/Hidden Markov Model* (CRF/HMM) e um método de caracterização de movimento chamado *Gesture Signature* que é computado usando fluxos ópticos a fim de descrever localização, velocidade e orientação do movimento global do gesto. Combinando a habilidade de modelagem do HMM e a habilidade discriminativa do CRF para classificar 30 vocabulários de 8 a 15 gestos dinâmicos em cada.

Utilizando uma calibração e retificação estéreo para obter imagens retificadas e correspondência estéreo para obter o mapa de profundidade de imagens de duas câmeras diferentes, o trabalho de Laskar et al. (2015) extraiu um mapa baseado em disparidade do movimento do centroide e a mudança na sua intensidade para serem utilizados como características em um classificador CRF para o reconhecimento dos 10 gestos estáticos.

Somando 12 trabalhos, os estudos que buscam o reconhecimento de gestos dinâmicos utilizando sensores (KUMAR et al., 2017; BARROS et al., 2017; KUMAR et al., 2016; LIM; TAN; TAN, 2016a; LASKAR et al., 2015; ESCOBEDO-CARDENAS; CAMARA-CHAVEZ, 2015; ALMEIDA; GUIMARÃES; RAMÍREZ, 2014; JANGYODSUK; CONLY; ATHITSOS, 2014; HUANG et al., 2014; BRASHEAR et al., 2006; HERNANDEZ-REBOLLAR, 2005; JIANG; YAO; YAO, 2004) são os mais numerosos da revisão sistemática.

Nesta classe, (BRASHEAR et al., 2006) e (HUANG et al., 2014) tratam do reconhecimento de palavras ou expressões em vez de letras ou dígitos, que são o tipo mais presente nos trabalhos da revisão. Brashear et al. (2006) apresenta o sistema CopyCat que é um jogo digital que utiliza o reconhecimento de expressões dinâmicas pelo computador para desenvolver o conhecimento sobre *American Sign Language* utilizado classificação Bayesiana para gerar uma mascara binária e então classificar utilizando um HMM.

Em (HUANG et al., 2014), imagens RGBD obtidas com o sensor Kinect foram utilizadas para extrair esqueletos 3D e a técnica HOG para formato da mão como características espaciais e geométricas. Além destas, as pirâmides do esqueleto 3D e do resultado da HOG são utilizadas como característica temporal utilizando o *Histogram of Oriented Displacements* (HOD) para classificar 100 sinais dinâmicos de palavras.

Utilizando uma combinação de dados obtidos através do sensor Kinect e do Leap Motion, (KUMAR et al., 2016) e (KUMAR et al., 2017) propuseram uma fusão multi sensores para o reconhecimento de sinais dinâmicos. O primeiro apresenta um *framework* que identifica os sinais usando *Coupled Hidden Markov Model* (CHMM) e que utiliza

como classificador a SVM para classificar 50 sinais dinâmicos. O segundo buscou uma melhor acurácia usando características como a direção e posição dos dedos juntamente com os classificadores HMM e *Bidirectional Long Short-Term Memory Neural Network* (BLSTMNN) buscando classificar 25 sinais dinâmicos.

Buscando identificar gestos dinâmicos com imagens de Kinect, (ESCOBEDO-CARDENAS; CAMARA-CHAVEZ, 2015) integrou informações globais e locais dos gestos dinâmicos. O esqueleto 3D, a trajetória do movimento em coordenadas esférias, *Scale Invariant Feature Transform* (SIFT) e *Bag of Words* (BOW) foram utilizados para extrair características dos *frames* dos vídeos, alcançaram resultados promissores nas duas bases de 20 gestos em que o método foi testado.

Dos trabalhos de gestos dinâmicos, apenas um utilizou informações de imagens de intensidade luminosa e dados de sensores. Em (LIANG; GUIXI; HONGYAN, 2015) regiões de interesse são extraídas combinando imagens de intensidade luminosa e o mapa de profundidade. Para estabelecer um modelo para gestos estáticos, a SVM é utilizada com o resultado da aplicação dos momentos invariantes de Hu nos mapas de profundidade e do HOG nas imagens coloridas. Finalmente uma HMM é utilizada para manipular as entradas contínuas de dados para o treino e reconhecimento.

Para fins de resumo, a Tabela 2 apresenta todos os trabalhos de reconhecimento de sinais dinâmicos utilizados nessa revisão sistemática, em termos de base e acurácia obtidos.

Tabela 2 – Tabela de resumo dos trabalho de reconhecimento de sinais dinâmicos, ordenados por ano de publicação e acurácia.

Trabalho	Acurácia (%)	#Classes	#Indivíduos
(BARROS et al., 2017)	97	9 gestos	900 vídeos
(KUMAR et al., 2017)	94,55	50 sinais	7500 vídeos
(BELGACEM; CHATELAIN; PAQUET, 2017)	-	-	750 vídeos
(LIM; TAN; TAN, 2016a)	93,33	50 sinais	483 vídeos
(KUMAR et al., 2016)	90,80	25 sinais	2000 vídeos
(LI; ZHOU; LEE, 2016)	87,4	510 sinais	-
(LIM; TAN; TAN, 2016b)	87,33	50 sinais	483 vídeos
(ESCOBEDO-CARDENAS; CAMARA-CHAVEZ, 2015)	98,28	18 sinais	360 vídeos
(LIANG; GUIXI; HONGYAN, 2015)	93,78	5 gestos	1500 vídeos
(LASKAR et al., 2015)	88	10 sinais	-
(JANGYODSUK; CONLY; ATHITSOS, 2014)	93,38	1113 sinais	3339 vídeos
(HUANG et al., 2014)	89,9	100 sinais	900 vídeos
(ALMEIDA; GUIMARÃES; RAMÍREZ, 2014)	'acima de 80'	34 sinais	170 vídeos
(MADEO et al., 2012)	84,3	26 letras	-
(MADEO et al., 2010)	85,8	26 letras	-
(DIAS et al., 2009)	98	15 gestos	360 vídeos
(BRAGATTO; RUAS; LAMAR, 2006)	99,2	26 letras	-
(BRASHEAR et al., 2006)	93,39	22 sinais	1959 vídeos
(HERNANDEZ-REBOLLAR, 2005)	99,3	22 sinais	-
(JIANG; YAO; YAO, 2004)	-	-	-

2.3 Outros Trabalhos

Ao longo do estudo, alguns trabalhos relevantes também foram encontrados à parte do estudo dirigido. A grande maioria busca extrair características modeladas artificialmente através de redes neurais, porém um dos que foge a esta regra é (MARIN; DOMINIO; ZANUTTIGH, 2014), que utiliza características extraídas do sensor Kinect e do Leap Motion que, baseado na posição e orientação dos dedos, são utilizadas como entrada em uma SVM multi-classe a fim de reconhecer 10 classes de gestos da *American Manual Alphabet* presentes em uma base pública apresentada neste mesmo trabalho.

Outro trabalho, no mesmo sentido, foi proposto por Ronchetti (2017) através de uma rede *Self Organized Map* (SOM) Probabilística, a ProbSOM (ESTREBOU; LANZARINI; HASPERUÉ, 2010), para reconhecer gestos da LSA64 (RONCHETTI et al., 2016). A ProbSOM permite inferir estatisticamente agrupando classes de gestos semelhantes e em seguida determinando quais as características mais importantes para a discriminação de cada gesto.

Recentes avanços na capacidade de processamento e o aumento da quantidade de grandes bases de dados estão permitindo a aplicação de técnicas de aprendizado de máquina que, até então, eram quase impraticáveis dado seus altos custos computacionais e em suas necessidades por grandes quantidades de dados. A *deep learning* é uma das técnicas mais promissoras, sendo utilizada com sucesso no reconhecimento automático de fala (GRAVES; MOHAMED; HINTON, 2013; HINTON et al., 2012; DAHL et al., 2012), sistemas de recomendação (WANG; WANG; YEUNG, 2015; WANG; WANG, 2014) e reconhecimento de imagens (HE et al., 2016; SIMONYAN; ZISSERMAN, 2014; CIREŞAN et al., 2013; CIREGAN; MEIER; SCHMIDHUBER, 2012).

Quando se trata de reconhecimento de gestos dinâmicos, o trabalho de Pigou et al. (2014) emprega duas CNNs, cuja entrada são os vídeos em nível de cinza e de profundidade, originais da base CLAP14. A primeira rede utiliza os vídeos originais, enquanto a segunda corta cada frame de modo que o resultado contenha somente a mão do usuário. Ao final, os vetores de característica são combinados para então classifica-los.

Ainda utilizando CNN para o reconhecimento de gestos, o trabalho de Huang et al. (2015) utilizou uma CNN com convoluções em três dimensões (3D CNN). Cada vídeo foi dividido em 5, denominados: color-R, color-G, color-B, profundidade e esqueleto do corpo, cada um com 9 frames, sendo os 3 primeiros referentes ao canal de cor do vídeo original. A base utilizada no trabalho é própria e contém 25 classes de gestos.

Apesar de não tratar sobre o reconhecimento de LS e sim de gestos em geral, vale citar o trabalho de Molchanov et al. (2015a), que utiliza uma técnica para ampliar a base de vídeos através de algumas transformações espaciais e deformações além de duas redes 3D CNN em paralelo. A primeira tem como entrada imagens da base VIVA (OHN-

BAR; TRIVEDI, 2014) e a segunda utiliza as mesmas imagens da primeira porém reduz significativamente as dimensões das imagens. As probabilidades resultantes da camada final de cada rede são combinadas através de probabilidade condicional para, então, classifica-las em uma das 19 classes.

3 Fundamentação Teórica

Este capítulo apresenta os conceitos teóricos utilizados neste trabalho. São abordados os conceitos: processamento digital de imagens, amostragem de vídeo, *deep learning*, convolução, CNN, convolução 3D, 3D CNN, MaxPooling, funções de ativação, otimização e função de aprendizado.

3.1 *Deep learning*

Segundo LeCun, Bengio e Hinton (2015) *deep learning* são métodos que permitem que modelos computacionais compostos de múltiplas camadas de processamento aprendam representações de dados com múltiplos níveis de abstração. Esse aprendizado é feito através da descoberta de estruturas intrincadas presentes em grandes volumes de dados, usando algoritmos de *backpropagation* para indicar como a máquina deve mudar seus parâmetros internos de forma que a camada atual aprenda as representações da camada anterior.

Deep learning é um tipo de Aprendizado de Máquina, que por sua vez é um ramo da Inteligência Artificial (IA). No início do campo da IA foram abordados e solucionados problemas que eram intelectualmente difíceis para humanos mas relativamente fáceis para computadores, ou seja, problemas que podem ser descritos por uma lista de regras matemáticas. O verdadeiro desafio da IA está em tarefas que são facilmente solucionadas por seres humanos mas que são difíceis de descrever formalmente, como reconhecimento de fala, faces ou imagens (GOODFELLOW; BENGIO; COURVILLE, 2016).

Algumas das soluções para esse tipo mais intuitivo de problemas tinham como cerne características formalmente especificadas por operadores humanos, especificando todo o conhecimento que o computador necessita (HUANG et al., 2014; BELGACEM; CHATELAIN; PAQUET, 2017; LASKAR et al., 2015; LIANG; GUIXI; HONGYAN, 2015). O problema com esse tipo de abordagem é a dificuldade em saber que tipo de características devem ser extraídas (GOODFELLOW; BENGIO; COURVILLE, 2016).

As abordagens de *deep learning* extinguem a necessidade de definir formalmente quais características o classificador deve usar através de uma representação hierárquica de conceitos. O conhecimento é reunido por experiência, permitindo que o computador aprenda conceitos complicados através de conceitos mais simples (GOODFELLOW; BENGIO; COURVILLE, 2016). Estes métodos melhoraram excepcionalmente o estado da arte em reconhecimento de fala, reconhecimento de objetos visuais, detecção de objetos e muitos outros domínios tais como descoberta de novos medicamentos e genômica (LECUN; BENGIO; HINTON, 2015).

Os modelos de *deep learning* também são conhecidos como redes neurais artificiais pois seus primeiros algoritmos foram intencionalmente construídos para representar o aprendizado biológico, ou seja, modelos de como o aprendizado acontece no cérebro.

A arquitetura de uma rede neural se refere a toda a estrutura da rede: a quantidade de unidades e como essas unidades devem se conectar. A maioria das redes neurais são organizadas em grupos chamados de camadas, compostas por unidades denominadas neurônios. A maioria das redes neurais organiza essas camadas em uma estrutura em cadeia, com cada camada sendo uma função da camada que a antecede (GOODFELLOW; BENGIO; COURVILLE, 2016). Nestas arquiteturas em cadeia, as principais decisões arquiteturais são a escolha da quantidade de camadas da rede e a largura (quantidade de neurônios) de cada camada.

3.2 *Nearest Neighborhood Interpolation*

Em estatística, amostragem é o processo de obtenção de amostras, que são uma pequena parte de uma população (SPIEGEL; SCHILLER; SRINIVASAN, 2016). No contexto deste trabalho, amostrar os vídeos trata-se de criar ou remover *frames* (cada uma das imagens presentes em um vídeo) dada uma finalidade. Há várias maneiras de amostrar um vídeo, como a *Temporal Augmentation* (MOLCHANOV et al., 2015a) e a *Nearest Neighborhood Interpolation* (NNI) (MOLCHANOV et al., 2015b), por exemplo. Neste trabalho, utilizamos a NNI a fim de tornar constante o tamanho dos vídeos, ou seja, igualar a quantidade de *frames* presentes em cada vídeo. Esta etapa é necessária pois é um requisito do classificador que todas os valores da entrada tenham o mesmo tamanho.

O *Nearest Neighborhood Interpolation* trata-se de um método para amostrar os vídeos removendo ou repetindo-os, dependendo se é preciso diminuir ou aumentar a quantidade de *frames* presentes em cada um dos vídeos. A remoção/repetição se dá visando a maior diversidade de *frames* possível, ou seja, que a remoção/repetição aconteça em intervalos iguais. Por exemplo, se desejamos que os vídeos tenham 60 *frames* e a quantidade original é 80, devemos remover cada quarto *frame*. Caso o tamanho do vídeo seja 45 *frames*, devemos clonar cada terceiro *frame* (MOLCHANOV et al., 2015b).

Supondo que v é o vetor contendo os n *frames* do vídeo original e m é a quantidade de *frames* que desejamos que o vídeo normalizado possua. Podemos calcular o vetor c, que contém quais as posições dos *frames* de v que devem estar no vídeo normalizado da seguinte forma:

$$c = \lceil (\frac{n}{m}) * [1, 2, 3, \ldots, m] \rceil \tag{3.1}$$

onde $\lceil x \rceil$ representa a função teto de x e $*$ representa uma multiplicação por escalar.

Esta técnica será utilizada na etapa de pré-processamento, a fim de adequar os vídeos aos requisitos da técnica de aprendizado que utilizaremos. Para exemplificar, considere um vídeo contendo 12 *frames* e que desejamos normaliza-lo para 7, então c deve ser calculado da seguinte forma:

$$c = \lceil \frac{12}{7} * [1, 2, 3, 4, 5, 6, 7] \rceil$$
$$c = [2, 4, 6, 7, 9, 11, 12]$$

3.3 *Convolutional Neural Networks*

Convolutional Neural Networks são um tipo específico de rede neural de *deep learning* modelada para processar dados em formato de múltiplos vetores como, por exemplo, uma imagem colorida composta por três vetores 2D, cada um correspondendo a intensidade dos *pixels* (unidade em uma matriz discreta) em 3 canais de cores diferentes. Exemplos incluem séries temporais, que podem ser pensadas como múltiplos vetores 1D se forem consideradas amostras em intervalos de tempo regulares, além de imagens e vídeos, que podem ser categorizados como múltiplos vetores 2D e 3D, respectivamente (LECUN; BENGIO; HINTON, 2015; GOODFELLOW; BENGIO; COURVILLE, 2016).

Esta rede possui este nome pois utiliza uma operação matemática denominada convolução. Convolução é um tipo especializado de operação linear, dessa forma podemos pensar em CNNs como redes neurais que usam convolução no lugar a multiplicação matricial em pelo menos uma das camadas (GOODFELLOW; BENGIO; COURVILLE, 2016). As camadas que utilizam a operação de convolução são denominadas camadas de convolução.

As CNNs são um tipo de aprendizado supervisionado, ou seja, associam as entradas com as saídas dado um conjunto de treinamento. Essa associação é feita através da modificação de parâmetros internos presentes em uma região intermediária composta por várias camadas, que são comumente designadas como camadas escondidas pois seus valores não estão presentes nos dados; na verdade o modelo deve determinar que conceitos são úteis para explicar as relações entre os dados observados (GOODFELLOW; BENGIO; COURVILLE, 2016).

Em uma CNN, a arquitetura típica da camada escondida é estruturada como uma série de estágios. Os primeiros estágios são compostos por dois tipos de camadas: camadas de convolução e camadas de *pooling* (LECUN; BENGIO; HINTON, 2015). Estes primeiros estágios são comumente chamados de estágios de extração de características, pois são neles que a rede aprende quais são os padrões mais importantes para a classificação posterior.

3.3.1 Convolução 2D

No contexto de redes convolucionais (LECUN et al., 1998), a convolução mais comum é a convolução 2D, que é calculada nas camadas convolucionais onde as características são extraídas de vizinhanças locais presentes nos resultados das camadas anteriores. Matematicamente a convolução 2D em um ponto (x, y), considerando um núcleo de convolução de tamanho (P, Q) pode ser expressa da seguinte forma:

$$O_{xy} = \sum_{p=0}^{P-1} \sum_{q=0}^{Q-1} K_{pq} I_{(x+p)(y+q)} \tag{3.2}$$

onde O é o resultado da convolução 2D, (p, q) são as coordenadas dos *pixels* do núcleo de convolução K e I é a matriz de entradas.

Ou seja, cada *pixel* em uma posição (x, y) é multiplicado por um *pixel* em uma posição equivalente no núcleo de convolução. Ao final, cada uma destas multiplicações é somada e então este valor é armazenado na matriz resultante.

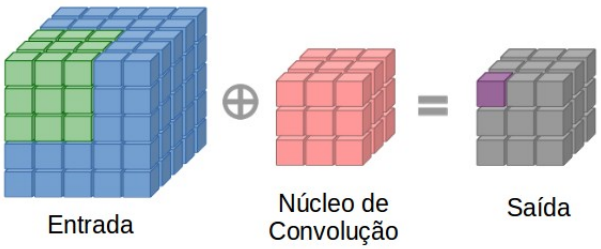
3.3.2 3D CNN

A extensão das CNNs para 3D, chamadas 3D CNNs (JI et al., 2013) se diferenciam das redes CNN pois utilizam ao menos uma convolução 3D. Essas convoluções podem, além de extrair informações espaciais das matrizes como as convoluções 2D, extrair informações presentes entre as matrizes consecutivas. Este fato nos permite mapear tanto informações espaciais de objetos 3D como informações temporais de um conjunto de imagens sequenciais.

3.3.2.1 Convolução 3D

A convolução 2D pode ser estendida para três dimensões, na qual não consideramos mais matrizes e sim volumes. Uma convolução 3D (JI et al., 2013) é uma operação matemática em que cada *voxel* (unidade em um volume discreto) presente no volume de entradas é multiplicado pelo *voxel* na posição equivalente do núcleo de convolução. Ao final, cada um destes resultados é somado e então são adicionados a saída. Na Figura 3 é possível observar a representação da operação de convolução 3D, onde os *voxels* em destaque na Entrada são multiplicados com o seus respectivos *voxels* no Núcleo de convolução, após isso todos são somados, gerando o *voxel* em destaque presente na Saída.

Figura 3 – Representação de uma operação de convolução 3D.



Fonte: Elaborada pelo autor.

Considerando que as coordenadas do volume da entrada são dadas por (x, y, z) e o núcleo de convolução tem tamanho (P, Q, R) a operação de convolução 3D pode ser definida matematicamente como:

$$O_{xyz} = \sum_{p=0}^{P-1} \sum_{q=0}^{Q-1} \sum_{r=0}^{R-1} K_{pqr} I_{(x+p)(y+q)(z+r)} \tag{3.3}$$

onde O é o resultado da convolução, (p, q, r) são as coordenadas dos *voxels* do núcleo de convolução K e I é o volume da entrada. É essa a operação utilizada pelas camadas de convoluções 3D presentes nas 3D CNN.

3.3.2.2 *Pooling*

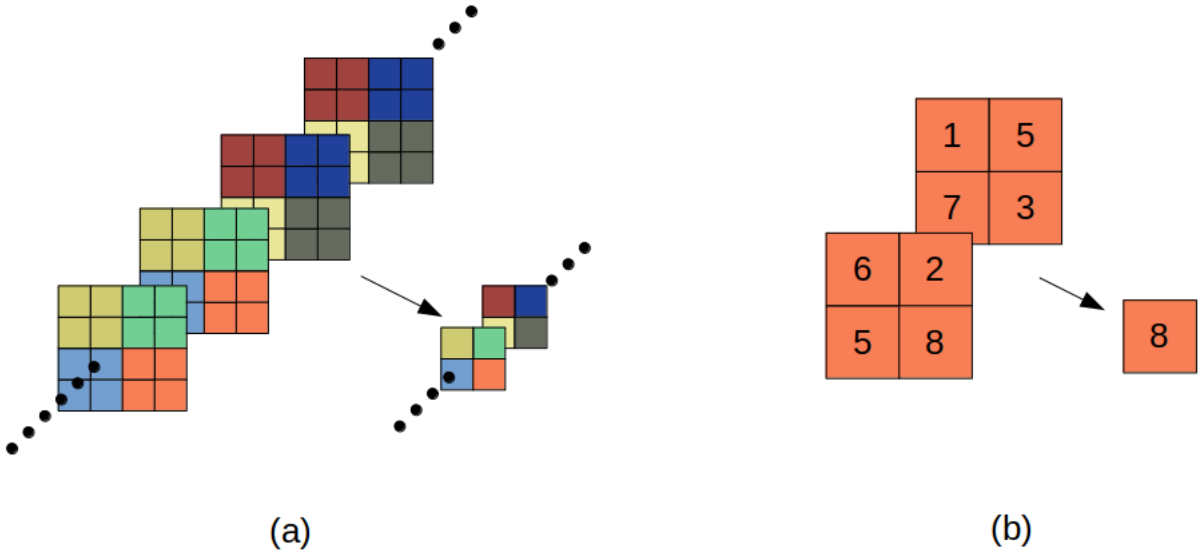
A camada de *pooling*, apesar de não ser necessária em uma CNN, é empregada em quase todas (OYEDOTUN; KHASHMAN, 2017; PIGOU et al., 2014; HUANG et al., 2015). Normalmente, o cálculo de convolução é feito e uma função de ativação é aplicada ao resultado deste cálculo, sendo o próximo estágio normalmente uma camada de *pooling*. A camada de *pooling* é responsável por transformar uma determinada posição da entrada em um sumário estatístico dos valores vizinhos a esta posição (GOODFELLOW; BENGIO; COURVILLE, 2016).

Uma típica camada de *pooling* que calcula o máximo valor de uma vizinhança de um tensor é denominada MaxPooling. Esta função, bem como a convolução 2D, pode ser facilmente estendida para o domínio tridimensional.

A MaxPooling 3D é a extensão da MaxPooling para o domínio 3D, ela também calcula o máximo valor de uma vizinhança, no entanto a vizinhança considerada é um volume de tamanho (L, Q, P). Além do tamanho da vizinhança a ser considerada, há também um outro parâmetro chamado *stride*. O *stride* representa qual o incremento deve ser feito na posição atual para que o cálculo do próximo valor seja realizado.

O *pooling* de unidades adjacentes pode ser feito com o deslocamento (*stride*) de mais de uma linha, coluna ou profundidade, reduzindo a dimensão da entrada e

Figura 4 – Exemplo do cálculo de MaxPooling. (a) Representação das vizinhanças e do *stride*. (b) Resultado gráfico da função de valor máximo.



Fonte: Elaborada pelo autor.

criando invariância a pequenos deslocamentos e distorções, que pode aumentar o poder de generalização da rede (LECUN; BENGIO; HINTON, 2015).

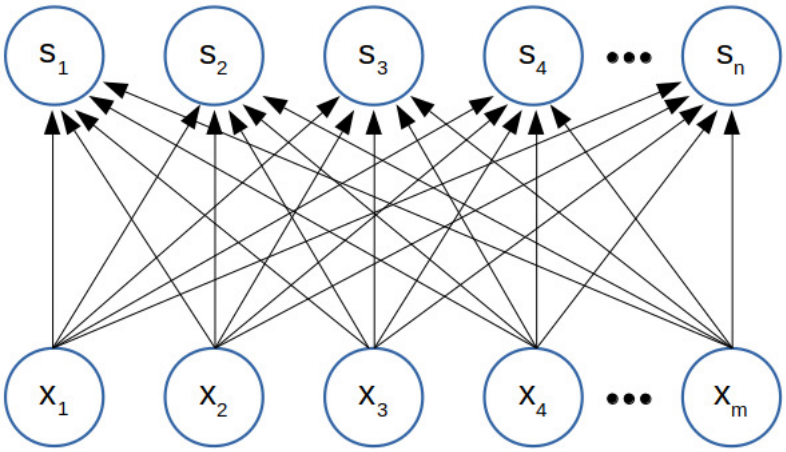
Na Figura 4 temos um exemplo da aplicação do MaxPooling 3D. Neste exemplo temos uma vizinhança de volume $(2, 2, 2)$ e *stride* $(2, 2, 2)$, não permitindo a sobreposição das regiões. Note que na Figura 4a o *stride* afeta como a posição atual é incrementada. A Figura 4b apresenta o cálculo de valor máximo entre todos os vizinhos, utilizado pela camada MaxPooling.

3.3.3 *Fully Connected*

Após passar pelo estágio de extração de características, a rede deve classificar as características extraídas em uma das classes do domínio. A esses últimos estágios atribuímos o nome de estágios de classificação. A camada mais utilizada para compor os estágios de classificação são as camadas *Fully Connected* (FC).

O nome desta camada está intimamente ligado a como seus neurônios são conectados, pois cada um deles está ligado a todos os neurônios da camada posterior. A Figura 5 representa esta ligação, onde cada uma das ligações representam um valor, o conjunto destes valores são chamados de pesos. Se considerarmos x as entradas e w os pesos da camada FC, os resultados s são dados pela multiplicação matricial entre a entrada e os pesos da rede.

Figura 5 – Representação gráfica de entre as ligações de uma camada FC.



Fonte: Elaborada pelo autor.

3.4 Função de Ativação

As funções de ativação são funções matemáticas aplicadas a saída de cada camada. O principal objetivo das funções de ativação é adicionar não-linearidade aos cálculos realizados pela rede neural, cujo conceito está intimamente ligado a ativação de um neurônio biológico. Uma das funções de ativação mais utilizada atualmente é a ReLU, que demonstra ser superior a funções previamente estabelecidas, como a sigmoide e a tangente hiperbólica (GOODFELLOW; BENGIO; COURVILLE, 2016).

3.4.1 ReLU

A *Rectified Logical Unit* (ReLU) (GLOROT; BORDES; BENGIO, 2011) é uma função de ativação definida como a parte positiva da entrada. Caso o neurônio esteja ativo, a função assume o valor da entrada, caso contrário, assume valor zero. Matematicamente a ReLU é definida como:

$$ReLU(x) = max(0, x) \tag{3.4}$$

As funções de ativação tem como objetivo adicionar não-linearidade às arquiteturas de rede neurais. A ReLU fornece esta característica através dos caminhos que se formam de acordo com os neurônios ativos (GLOROT; BORDES; BENGIO, 2011).

3.4.2 Softmax

A função Softmax pode ser vista como uma generalização da função sigmoide para múltiplos valores. Esta é utilizada para representar a distribuição de probabilidade sobre n classes diferentes. É mais frequentemente utilizada na saída do classificador, porém

também pode ser utilizada dentro das camadas escondidas. Formalmente a Softmax é definida de acordo com a Equação 3.5 (GOODFELLOW; BENGIO; COURVILLE, 2016).

$$softmax(x)_i = \frac{e^{x_i}}{\sum_j e(x_j)} \tag{3.5}$$

onde x_i é a entrada no i-ésimo neurônio e x_j são todos os outros valores.

3.5 Função Objetivo

Um aspecto importante quando estamos utilizando uma rede neural artificial é definir a função objetivo. A função objetivo é a função que o otimizador deve minimizar/maximizar a fim de reduzir a diferença entre os resultados calculados pela rede e os valores esperados. Em processos de otimização, geralmente buscamos minimizar a função objetivo. Esta minimização é guiada através do otimizador, que durante o treinamento dimensiona o ajuste dos pesos da rede visando selecionar as características mais relevantes para a classificação.

Em teoria da informação, a *cross entropy* consiste em calcular a quantidade média de bits necessários para identificar um evento de uma distribuição verdadeira p usando um esquema de codificação baseado na distribuição q. Essa função pode ser usada para definir a função objetivo de uma rede neural, onde distribuição verdadeira p_i é o valor esperado, e a distribuição codificada q_i é o valor calculado do modelo atual. Formalmente a *cross entropy* é apresentada na Equação 3.6 (HINTON; SALAKHUTDINOV, 2006).

$$\mathcal{L}(\omega) = -\frac{1}{n}\sum_{i=1}^{n}\left[p_i \log\left(q_i\right) + \left(1 - p_i\right)\log\left(1 - q_i\right)\right] \tag{3.6}$$

onde ω são os pesos da rede, n é a quantidade de observações, p_i e q_i são o valor esperado e o valor calculado da i-ésima observação, respectivamente.

Minimizar a *cross entropy* significa reduzir a diferença entre a distribuição verdadeira, classes reais, e a distribuição codificada, classes calculadas, visando aproximar o modelo aos indivíduos do treino, generalizando-o o suficiente para que seja capaz de ser capaz de representar os indivíduos do teste de maneira satisfatória.

3.6 RMSProp

A maioria dos algoritmos de *deep learning* envolve algum tipo de otimização. Otimização se refere a tarefa de minimizar ou maximizar a função objetivo. Quando tentamos minimiza-la também atribuímos o termo função de custo e cujo valores denominamos custos.

Um dos grandes problemas quando utilizamos otimizadores é estimar a taxa de aprendizado, pois esta afeta significativamente a performance do modelo. O algoritmo de *momentum* (POLYAK, 1964) reduz esse problema, mas o faz ao introduzir outro hiper parâmetro. Os custos são frequentemente sensíveis para algumas direções do espaço de parâmetros e insensível a outras. Se supormos que as sensibilidades as direções são, de alguma forma, alinhadas a um eixo, faz sentido usar uma taxa de aprendizado única para cada um dos parâmetros e automaticamente adaptar esses valores ao longo do aprendizado (GOODFELLOW; BENGIO; COURVILLE, 2016).

O *Root Mean Square Propagation* (RMSProp) (TIELEMAN; HINTON, 2012) é um otimizador que adapta individualmente as taxas de aprendizado de todos os parâmetros do modelo ao dimensiona-los através da acumulação do gradiente através de uma média móvel ponderada exponencialmente. O objetivo dessa média é descartar os valores mais antigos a fim de converter mais rapidamente após encontrar um hiperplano convexo.

Este algoritmo é uma estratégia de minimização do erro, baseada na robustez do RProp (RIEDMILLER; BRAUN, 1993), a eficiência dos *mini-batches* bem como um balanceamento efetivo dos mesmos. A utilização dos *mini-batches* se dá através da utilização de pequenos subconjuntos aleatoriamente escolhidos da base de treino para calcular o gradiente a cada iteração da rede.

Em uma rede cujos pesos iniciais são dados por θ, taxa de aprendizado ϵ, taxa de decaimento ρ e uma constante pequena δ, usada para estabilizar a divisão por números pequenos. Supondo que aleatoriamente m indivíduos do conjunto de treino $x^{(1)}, x^{(2)}, ..., x^{(m)}$ são selecionados com suas respectivas classes $y^{(i)}$. O gradiente pode ser calculado como:

$$g = \frac{1}{m} \nabla_\theta \sum_i \mathcal{L}(f(x^{(i)}; \theta), y^{(i)}) \tag{3.7}$$

onde g é o gradiente, f é a saída calculada quando a entrada é $x^{(i)}$, os pesos da rede são θ e $\mathcal{L}$ é a função que calcula o custo por cada indivíduo $x^{(i)}$.

Após calcular o gradiente, deve-se adiciona-lo ao gradiente acumulado, formalmente:

$$r = \rho r + (1 - \rho) g \odot g \tag{3.8}$$

onde r é o gradiente acumulado e ρ é a taxa de decaimento que dimensiona a maior importância para os gradientes mais recentes. O operador $\odot$ é a multiplicação de Hadamard (HUANG, 2008), em que cada valor na primeira matriz é multiplicado pelo valor na posição correspondente na segunda matriz.

Os valores de atualização dos pesos são calculados a partir do gradiente acumulado

da seguinte forma:

$$\Delta\theta = -\frac{\epsilon}{\sqrt{\delta + r}} \odot g \tag{3.9}$$

onde $\Delta\theta$ são os valores de atualização dos pesos, e $\sqrt{\delta + r}$ é o fator que escala o ϵ para cada um dos pesos.

Após calcular todos os valores para atualização dos parâmetros basta adiciona-los aos pesos, ou seja, estamos atualizando os pesos na direção de g:

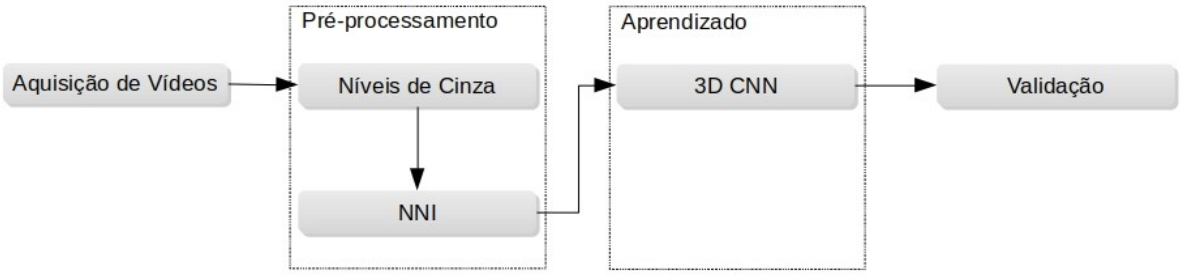
$$\theta = \theta + \Delta\theta \tag{3.10}$$

A cada iteração o gradiente é recalculado para um *batch* de m indivíduos, é este gradiente que atualiza os pesos da rede baseado no função de custo $\mathcal{L}$. Este processo ocorre enquanto o critério de parada da rede não for atingido.

4 Metodologia

Esse capítulo descreve a metodologia empregada para o reconhecimento de sinais, mais especificamente a arquitetura 3D CNN utilizada para o reconhecimento de sinais da LSA. A Figura 6 apresenta as etapas da metodologia, que são: Aquisição de Vídeos, Pré-processamento, Aprendizado e Validação, cada uma das etapas está detalhada nas seções seguintes.

Figura 6 – Metodologia Proposta.



Fonte: Elaborada pelo autor.

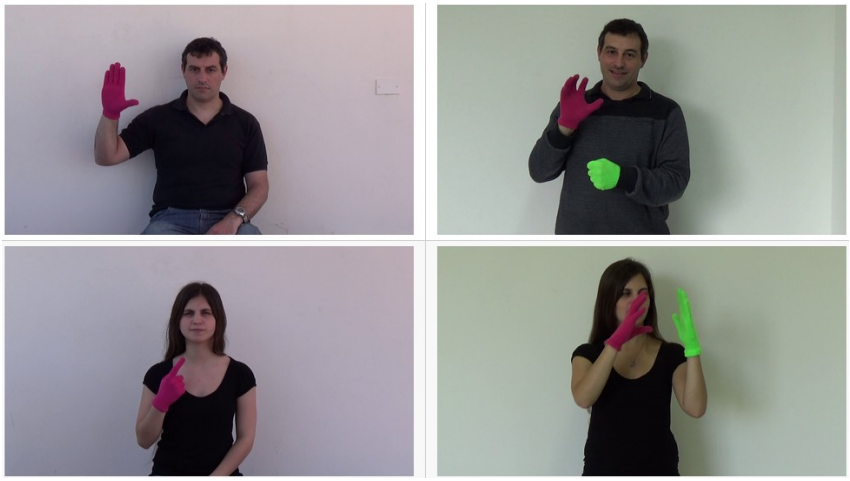
4.1 Aquisição de Vídeos

A base de vídeos LSA64 contém 3200 vídeos de sinais da LSA, onde 10 indivíduos não especialistas executam 5 repetições de 64 tipos diferentes de sinais usando uma ou ambas as mãos. Os sinais foram selecionados entre os mais comumente usados na LSA, incluindo verbos e substantivos (RONCHETTI et al., 2016). Alguns exemplos do conjunto de vídeos são apresentados na Figura 7.

A base de vídeos é composta de dois conjuntos. O primeiro contém 23 sinais que utilizam apenas uma mão e gravados em ambiente externo, com luz natural. O segundo contém 44 sinais, 22 com ambas as mãos e 19 com uma única mão e registrados em ambiente interno, com luz artificial, para fornecer diferenças na iluminação entre os sinais. Em ambos os conjuntos, os indivíduos vestiram trajes pretos e executaram os sinais em pé ou sentados, com uma parede branca ao fundo. Para simplificar problemas com segmentação, os indivíduos utilizaram luvas com cores fluorecentes. Cada sinal foi executado impondo poucas limitações, para aumentar a diversidade e o realismo da base de vídeos (RONCHETTI et al., 2016).

Todos os 10 indivíduos eram destros e não tinham experiência com línguas de sinais, foram instruídos em como executar cada um dos sinais durante as sessões de registro dos

Figura 7 – Exemplos de *frames* extraidos da LSA64.



Fonte: Modificada de (RONCHETTI et al., 2016).

vídeos, onde praticaram algumas vezes os sinais antes de serem filmados definitivamente. A câmera utilizada foi a mesma em ambos os conjuntos, uma Sony HDR-CX240. O tripé foi colocado a 2 metros de distância da parede, a uma altura de 1.5 metros. Marcações no chão foram utilizadas para indicar aos indivíduos onde deveriam se posicionar. A resolução dos vídeos é 1920x1080, a 60 *frames* por segundo (RONCHETTI et al., 2016).

A base de dados possui uma versão com cortes que é semelhante à versão original, porém cada vídeo foi temporalmente segmentado para que os quadros no início ou no final do vídeo, sem movimento nas mãos, fossem removidos. Esta versão com cortes da LSA64 foi a escolhida para todos os testes e construção do modelo.

4.2 Pré-processamento

Como estamos interessados em construir uma metodologia genérica para qualquer LS e a cor das luvas dos atores poderia facilitar o reconhecimento dos sinais de uma maneira artificial, optamos por converter os vídeos para níveis de cinza.

Cada video da LSA64 tem uma quantidade de *frames* diferente. Para normalizar a quantidade de *frames* por vídeo para 30 *frames* a técnica NNI é utilizada, pois remove ou repete *frames* dependendo se o video original contém mais ou menos que 30 *frames*, respectivamente.

Além da normalização da quantidade de *frames* por vídeo, a resolução espacial de cada vídeo foi reduzida para 80 x 45 *pixels*. Dessa forma nós reduzimos o custo computacional mantendo a proporção das imagens, de forma que o impacto nos resultados fosse mínimo.

4.3 Aprendizado

Após pré-processar os vídeos, eles são utilizados para treinar a rede 3D CNN a fim de ajustar os filtros da mesma para reconhecer padrões de cada um dos sinais. Idealmente devemos escolher a melhor configuração da 3D CNN. Com esse intuito, realizamos testes com variadas configurações. Definimos como otimizador o RMSprop dado seu sucesso em estudos relacionados (MNIH et al., 2016; KARPATHY; FEI-FEI, 2015), função objetivo *cross entropy* e funções de ativação são ReLU com exceção da última camada, na qual utilizamos a Softmax.

Cada um destes testes foi implementado utilizando a API Keras (CHOLLET et al., 2015) e com a biblioteca Theano (Theano Development Team, 2016) como *backend*. Todos os experimentos desta Seção foram repetidos três vezes, a fim de eximir fatores aleatórios com a utilização da média dos resultados. A proporção de treino/teste utilizada foi 80% para treino e 20% para teste.

Como fator de regularização do treinamento, definimos que a taxa de aprendizado começaria em $0,001$ e seria reduzida pela metade caso a função de custo não fosse reduzida em 5% nas próximas 20 iterações. O treino é finalizado caso a taxa de aprendizado caia pela quarta vez ou se o número de iterações exceder 150. Os outros parâmetros do optimizador foram utilizados com seu valor padrão (ρ de $0,9$ e taxa de decaimento igual a 0) (TIELEMAN; HINTON, 2012).

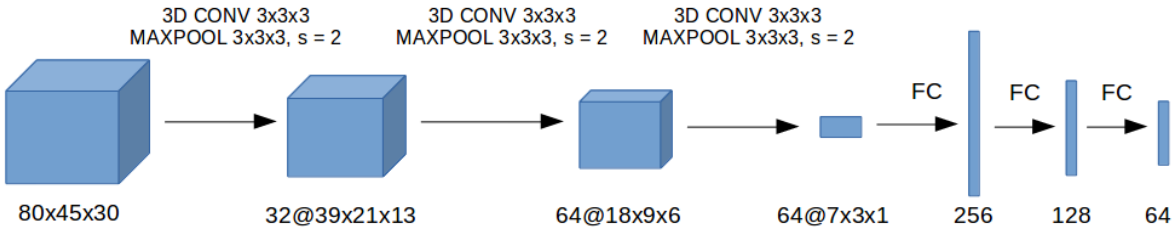
Para escolher a melhor quantidade de camadas de convolução 3D foram realizados testes com 2, 3 e 4 camadas de convolução 3D, cada uma acompanhada por uma MaxPooling 3D. Nestes testes utilizamos 32 filtros de tamanho $(3, 3, 3)$ em cada camada de convolução 3D e em cada camada MaxPooling 3D um núcleo de convolução de tamanho $(3, 3, 3)$ e *stride* $(2, 2, 2)$. No final das redes havia uma única camada FC com 64 neurônios.

Com a quantidade de camadas estabelecida, buscamos balizar a quantidade de filtros de tamanho $(3, 3, 3)$ em cada camada de convolução. Para isso fizemos testes com todas as combinações de 32 e 64 filtros, desde a utilização de 32 filtros em todas as camadas até os testes em que todas tinham 64 filtros, a quantidade de filtros foi limitada a 64 devido a problemas de memória insuficiente. Após definir a quantidade de filtros por camada, fizemos testes incrementais começando com uma única camada FC, até que em 4 camadas FC o resultado permaneceu praticamente o mesmo que 3 camadas, então optamos por utilizar apenas 3 camadas FC.

A arquitetura que obteve o melhor resultado está visualmente apresentada na Figura 8, em que optamos por representar as camadas de convolução 3D e MaxPooling 3D em uma só para simplificar o entendimento. A arquitetura contém 3 camadas de convolução 3D, cada uma seguida de uma camada de MaxPooling. Após a última camada de MaxPooling, a arquitetura possui 3 camadas FC de 256, 128 e 64 neurônios, nesta

sequência.

Figura 8 – Arquitetura 3D CNN proposta para o reconhecimento da LSA.



Fonte: Elaborada pelo autor.

A primeira convolução usou 32 filtros $(3, 3, 3)$ e é seguida por uma camada de MaxPooling 3D de núcleo de convolução $(3, 3, 3)$ e *strides* $(2, 2, 2)$. A segunda e a terceira contém 64 filtros $(3, 3, 3)$ e ambas são seguidas por uma camada MaxPooling 3D com núcleo de convolução $(3, 3, 3)$ e *strides* $(2, 2, 2)$. Todas as camadas apresentadas utilizam a ReLU como função de ativação. A última camada FC usa a função Softmax para interpretar as 64 saídas da rede, pois representam a probabilidade de uma dada classe pertencer a uma das 64 classes presentes na base.

4.4 Validação

O objetivo dessa etapa consiste em medir o desempenho e promover a validação dos resultados obtidos. Este trabalho utiliza as estatísticas: Acurácia, Sensibilidade e Precisão.

Estas estatísticas são calculadas com base na relação entre qual foi o sinal reconhecido pela 3D CNN com o seu tipo real do sinal. A matriz de confusão é uma maneira de visualizar a quantidade de classes que foram corretamente classificadas e onde houve erros de classificação. Neste trabalho convencionamos que as colunas representam as classes verdadeiras, enquanto as linhas representam as classes reconhecidas pela 3D CNN.

A Tabela 3 representa um exemplo de matriz de confusão. O reconhecedor classificou 10 indivíduos A corretamente, porém classificou incorretamente 2 como classe B e 3 como classe C. Além disso, temos que o reconhecedor classificou 2 indivíduos C como classe A. Então, para a classe A, temos 10 Verdadeiros Positivo (VP), 5 Falsos Negativo (FN) e 2 Falsos Positivo (FP). Os Verdadeiros Negativo (VN) representam a quantidade de indivíduos das outras classes (B e C, neste caso) que foram classificados corretamente, resultando em 26 indivíduos. De maneira similar, estes valores podem ser obtidos para as classes B e C.

A acurácia é definida na Equação 4.1 e corresponde à taxa de classificações corretas, sendo definida como a razão entre o número de indivíduos na amostra em estudo que foram

Tabela 3 – Exemplo de matriz de confusão com 3 classes

	A	B	C
A	10	0	2
B	2	14	1
C	3	1	12

classificados corretamente e o número total de indivíduos. Da maneira que é calculada, esta métrica representa a performance geral do teste, diferentemente da sensibilidade e precisão, explicadas mais abaixo.

$$Acurácia = \frac{VP + VN}{VP + VN + FP + FN} \tag{4.1}$$

A sensibilidade (Equação 4.2) define a proporção de Verdadeiros Positivos identificados nos testes, indicando o quão bom o teste é para identificar uma determinada classe A em relação a quantidade de indivíduos A presentes no teste.

$$Sensibilidade = \frac{VP}{VP + FN} \tag{4.2}$$

A precisão (Equação 4.3) representa a proporção de Verdadeiros Positivos quando consideramos a quantidade de Positivos classificados no teste, indicando o quão bom esse teste é para identificar uma determinada classe A baseado na quantidade de vezes que o reconhecedor classificou um indivíduo como A. Perceba que, assim como a sensibilidade, a precisão varia de classe para classe, portanto cada classe possui valores individuais para sensibilidade e precisão.

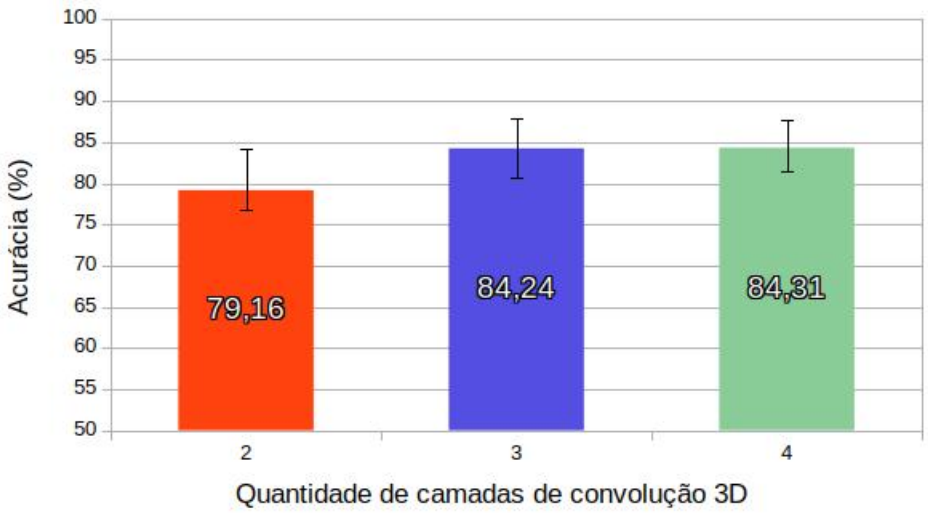
$$Precisão = \frac{VP}{VP + FP} \tag{4.3}$$

5 Resultados

Este capítulo apresenta os resultados produzidos pela metodologia apresentada e a análise dos mesmos. Os resultados estão organizados de acordo com a ordem apresentada no Capítulo 4.

Os primeiros experimentos realizados foram para descobrir a quantidade de camadas convolucionais a serem utilizadas na arquitetura 3D CNN. As médias das acurácias dos 3 experimentos para cada quantidade de camadas estão apresentadas na Figura 9.

Figura 9 – Resultados dos experimentos para cada quantidade de camadas de convolução 3D.



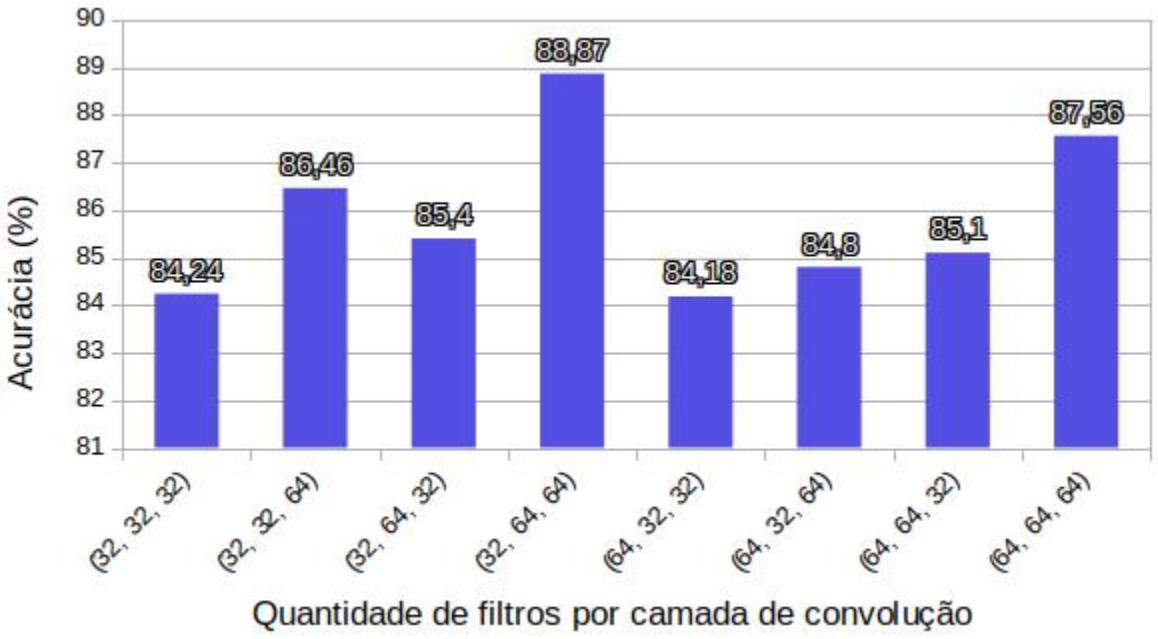
Fonte: Elaborada pelo autor.

A diferença na acurácia média quando utilizamos 2 e 3 camadas de convolução foi de 5,08 pontos percentuais (pp). Ao adicionar uma quarta camada, o resultado médio permaneceu praticamente o mesmo que obtivemos com 3 camadas, subindo apenas 0,07 pp, que não chega a ser considerado uma grande melhoria nos resultados, no entanto gera um aumento no custo computacional. Por esse motivo, optou-se por utilizar 3 camadas nos testes seguintes.

Nos testes anteriores, todas as camadas de convolução utilizavam 32 filtros. A fim de utilizar uma quantidade mais apropriada de filtros em cada camada, realizamos experimentos com todas as combinações de 32 e 64 filtros para cada camada. Os resultados destes testes são apresentados na Figura 10.

É possível perceber que há uma queda de desempenho quando um afinamento da topologia é criado, ou seja, quando a quantidade de filtros de uma camada é menor do

Figura 10 – Resultados dos experimentos para cada configuração de rede utilizando 32 ou 64 filtros em cada camada de convolução.

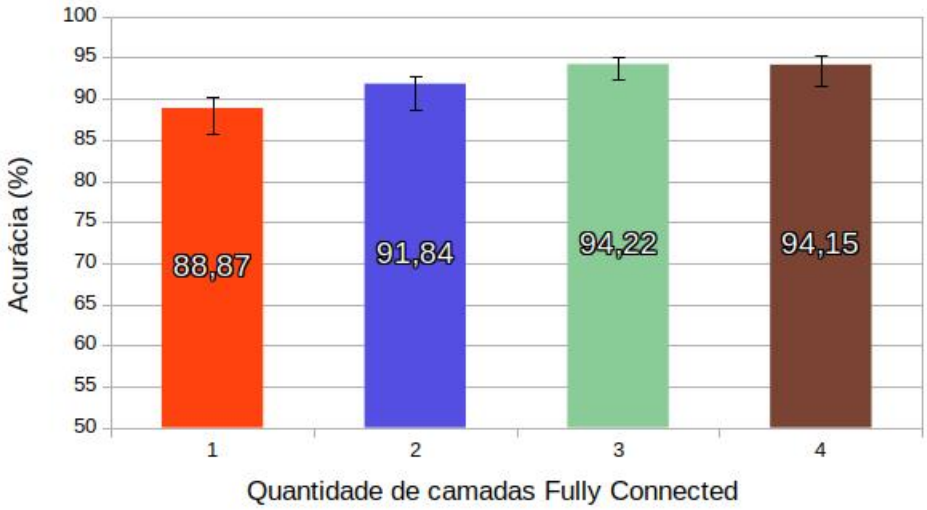


Fonte: Elaborada pelo autor.

que a quantidade de filtros da camada anterior. Este fato é verificado quando utilizamos as configurações $(32, 64, 32)$, $(64, 32, 32)$, $(64, 32, 64)$ e $(64, 64, 32)$, onde há uma redução na acurácia dos testes. A melhor configuração obtida através dos testes foi a $(32, 64, 64)$, que conseguiu um desempenho superior até mesmo da configuração $(64, 64, 64)$, que supostamente deveria apresentar os melhores resultados, obtendo a acurácia de $88,87\%$, 4,63 pp acima da acurácia do experimento inicial que utilizava apenas 32 filtros em todas as camadas.

Também realizamos testes onde 2, 3 e 4 camadas FC foram testadas na melhor configuração. A Figura 11 apresenta a média das acurácias obtidas nos testes utilizando 1, 2, 3 e 4 camadas FC. Adicionamos mais camadas FC até que a acurácia média não obtivesse uma melhora significativa nos resultados, fato que se deu ao adicionarmos a quarta camada, onde o resultado foi uma acurácia média menor quando comparada a acurácia média com 3 camadas.

Figura 11 – Resultados dos experimentos com 1, 2, 3 e 4 camadas FC.



Fonte: Elaborada pelo autor.

A Tabela 4 apresenta a matriz de confusão da soma dos 10 testes realizados com a melhor arquitetura encontrada em nossos experimentos (Figura 8), variando aleatoriamente os indivíduos utilizados para treino e teste em cada um destes.

Tabela 4 – Matriz de confusão dos 10 testes realizados.

1	73	0	0	0	0	0	0	0	0	0	0	0	3	0	3	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	115	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	56	25	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	15	133	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	90	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	3	0	1	0	81	0	3	0	0	0	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	97	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	98	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	99	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	87	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	1	0	0	0	0	0	105	0	0	0	0	2	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	105	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	90	0	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	3	0	2	0	0	0	108	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	3	2	0	114	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	72	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	103	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	2	0	0	0	2	1	0	0	0	0	0	87	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	1	0	2	0	0	0	0	0	0	0	0	0	0	95	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	12	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	3	106	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	97	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	91	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	5	0	0	0	0	0	0	0	2	0	1	0	0	0	0	0	0	3	0	0	0	0	99	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	102	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	106	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	66	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	118	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0
28	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	112	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	86	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	85	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	111	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	83	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	125	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	1	0	0	0
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	101	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	99	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	101	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0
39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	74	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0	0	0	0	0	0	0	0	0	0	74	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	97	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	83	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0
43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	104	0	0	0	0	0	0	2	0	0	0	2	1	0	0	0	0	0	0	0	0	0
44	0	0	1	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	93	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	84	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
46	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	74	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
47	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	4	128	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3
48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	91	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	3
49	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	73	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	111	0	0	0	0	0	0	0	2	0	0	0	0	0	0
51	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	105	0	0	0	0	0	0	0	0	0	0	0	0	0
52	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	68	0	0	0	0	0	0	0	0	0	0	0	6
53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	8	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	85	0	0	0	0	0	0	0	0	0	0	0
54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	0	2	0	0	0	88	0	0	0	0	0	0	0	0	0	0
55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	3	0	0	0	100	0	0	0	2	0	0	0	0	0
56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	75	1	0	0	0	0	0	0	0
57	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	70	2	0	0	0	0	0	0
58	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	99	0	0	0	0	0	0
59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	85	0	0	0	0	0
60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94	0	0	0	0
61	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	96	0	0	0
62	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	91	0	0
63	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	86	0
64	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	106

A partir da matriz de confusão calculamos as médias da sensibilidade e precisão para cada uma das classes, sendo estes valores apresentados na Tabela 5. Temos que as menores sensibilidades foram encontradas para os sinais 3, 4, 46, 59 e 48 e os menores valores de precisão em 3, 6, 49, 42 e 20. Neste conjunto das piores sensibilidades e precisão temos o par de sinais 3 e 4, Vermelho e Verde, respectivamente, que foram bastante confundidos pelo classificador.

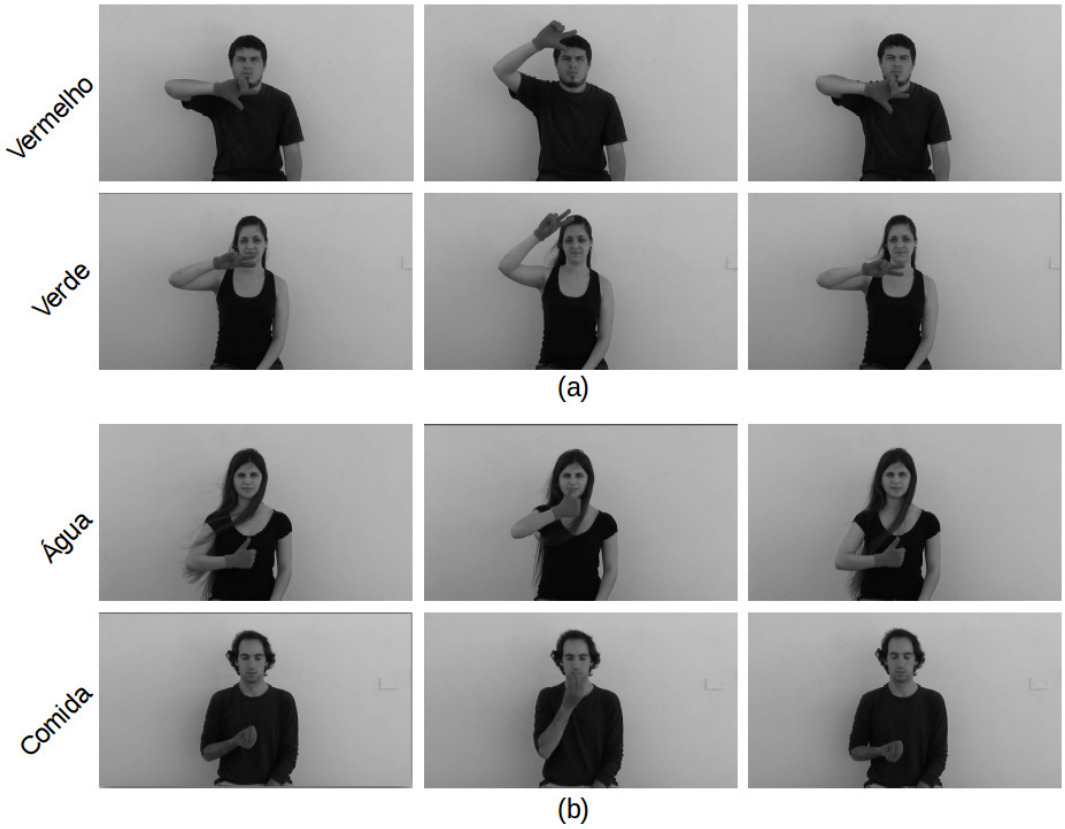
Tabela 5 – Média da Sensibilidade e Precisão para cada classe.

Classe	Sinal	Sensibilidade (%)	Precisão (%)
1	Comida	93,59	84,88
2	Opaco	87,12	97,46
3	Vermelho	77,78	66,67
4	Verde	80,61	88,67
5	Amarelo	96,77	100,00
6	Brilhante	98,78	81,82
7	Azul claro	97,98	100,00
8	Cores	90,74	100,00
9	Cor-de-rosa	98,02	100,00
10	Mulheres	93,55	96,67
11	Inimigo	97,22	96,33
12	Filho	93,75	98,13
13	Homem	94,74	95,74
14	Distante	99,08	93,91
15	Gaveta	87,69	95,80
16	Nascer	94,74	90,00
17	Aprender	94,50	100,00
18	Ligar	90,63	93,55
19	Espumadeira	87,16	96,94
20	Amargo	100,00	84,13
21	Doce	90,65	97,00
22	Leite	96,81	98,91
23	Água	92,52	90,00
24	Argentina	91,89	99,03
25	Uruguai	92,98	95,50
26	País	97,06	100,00
27	Onde	97,52	98,33
28	Último Nome	100,00	96,55
29	Zombar	87,72	100,00
30	Aniversário	93,48	100,00
31	Café da Manhã	97,70	100,00
32	Foto	100,00	99,11

33	Fome	98,81	93,26
34	Mapa	100,00	96,90
35	Moeda	100,00	100,00
36	Musica	97,06	98,02
37	Nave Espacial	98,06	100,00
38	Nenhum	90,91	94,34
39	Nome	100,00	85,06
40	Paciência	100,00	96,10
41	Perfume	97,00	97,00
42	Surdo	93,26	83,84
43	Armadilha	88,89	95,41
44	Arroz	97,89	93,00
45	Barbecue	94,38	100,00
46	Bombom	84,09	92,50
47	Chiclete	96,24	91,43
48	Espaguete	84,26	94,79
49	Iogurte	94,81	82,95
50	Aceitar	93,28	93,28
51	Agradecer	93,75	92,92
52	Desligar	97,14	88,31
53	Parecer	100,00	89,47
54	Pousar	95,65	90,72
55	Pegar	99,01	90,09
56	Ajudar	100,00	98,68
57	Dançar	98,59	92,11
58	Tomar banho	96,12	97,06
59	Comprar	84,16	96,59
60	Copiar	97,92	95,92
61	Correr	98,97	98,97
62	Perceber	100,00	90,10
63	Dar	98,85	92,47
64	Encontrar	88,33	92,98

Na Figura 12a temos a possível causa para os erros entre os sinais Vermelho e Verde. Nela temos 3 *frames* que representam os principais pontos de interesse em cada um dos gestos, sendo eles movimentos verticais que vão do queixo ao topo da cabeça, sendo o sinal das mãos representado por dois dedos à mostra. Uma situação semelhante ocorre nos sinais Água e Comida (gestos 1 e 23), representados na Figura 12(b) e cujo os gestos também são representados através de um movimento vertical, porém iniciando no centro do tórax, indo até o queixo e retornando a posição inicial.

Figura 12 – Exemplos de *frames* de sinais da LSA64. (a) Comparação entre *frames* dos sinais Vermelho e Verde. (b) Comparação entre os *frames* dos sinais Água e Comida.



Fonte: Elaborada pelo autor.

De forma similar, verificamos que os sinais 46, 59, 48, 6, 49, 42 e 20 também apresentavam semelhanças temporais com os sinais reconhecidos erroneamente pelo classificador. Apesar disso, notamos que uma proporção relativamente alta de sinais alcançaram o 100% de sensibilidade ou 100% de precisão. As classes de sinais 20, 28, 32, 34, 35, 39, 40, 53, 56 e 62, correspondentes a $15,6\%$ de toda a LSA64 obtiveram 100% de sensibilidade. Enquanto que $18,8\%$ da LSA64, relativa às classes de sinais 5, 7, 8, 9, 17, 26, 29, 30, 31, 35, 37 e 45, alcançaram 100% de precisão.

A única classe de sinal que obteve o 100% em sensibilidade e precisão foi a 35 (Moeda), ao analisar este gesto é notável que é um dos únicos em que movimentos de translação das mãos não são presentes, permitindo que o classificador tenha facilidade ao reconhecê-lo. A sequência de *frames* que representam a classe Moeda é apresentada na Figura 13.

Figura 13 – Classe Moeda.

Fonte: Elaborada pelo autor.

A acurácia média dentre todos os testes está apresentada na Tabela 6 juntamente com a acurácia de trabalhos relacionados. Além da acurácia também estão apresentados: a metodologia, base de dados e número de classes utilizados em cada trabalho.

Tabela 6 – Comparação com trabalhos relacionados, em ordem decrescente de acurácia e agrupados pela base de vídeos utilizada.

	Metodologia	Base	nClasses	Acc (%)
Proposta	**3D CNN**	**LSA64**	**64**	**94,22** $\pm$ 2,12
(MASOOD et al., 2018)	CNN + RNN	LSA64	46	95,21
(RONCHETTI, 2017)	ProbSOM	LSA64	64	91,7
(YANG; ZHU, 2017)	CNN + LSTM	WLSL	40	98,43
(HUANG et al., 2015)	3D CNN	Privada	25	94,2
(PIGOU et al., 2014)	CNN	CLAP14	20	91,7
(MARIN; DOMINIO; ZANUTTIGH, 2014)	Posição dos dedos	MKLM	10	91,3
(MOLCHANOV et al., 2015a)	3D CNN	VIVA	19	77,5

Quando comparamos os resultados com trabalhos que utilizam características extraídas através de técnicas manualmente desenvolvidas, notamos que há uma ligeira vantagem em nossa arquitetura em relação a metodologia apresentada por (MARIN; DOMINIO; ZANUTTIGH, 2014).

A arquitetura também foi promissora quando comparada a outros trabalhos que usam *deep learning*. O trabalho de (HUANG et al., 2015) foi o trabalho relacionado cujo resultado foi o mais próximo do obtido pela metodologia proposta. No entanto, devemos salientar que a quantidade de informações fornecidas à rede foi um grande diferencial, pois utilizou 5 vídeos por indivíduo, enquanto utilizamos apenas um vídeo em nível de cinza. Em (PIGOU et al., 2014), apesar de usar duas redes para reconhecimento, os resultados não superam os obtidos neste trabalho.

Utilizando a mesma base deste presente trabalho, temos (RONCHETTI, 2017) que obteve uma acurácia $2,52$ pp abaixo da nossa utilizando-se de uma ProbSOM, que não se mostrou tão eficiente ao classificar gestos com um mesmo sentido de movimento em comparação com a arquitetura proposta. Além deste trabalho, temos (MASOOD et al., 2018), que utilizou uma combinação de CNN para extração de características em conjunto com a *Recurrent Neural Network* (RNN) para classificação. Este obteve $95,21\%$

de acurácia, mas ao custo da remoção de classes de difícil reconhecimento como os sinais: Verde-Vermelho e Água-Comida, que foram classes que foram mantidas em nosso trabalho e geram impacto à avaliação de nossa arquitetura.

Em (YANG; ZHU, 2017) temos a utilização de uma rede CNN em conjunto com os estados ocultos de uma *Long Short-Term Memory* (LSTM) que caracterizam a que classe o próximo *frame* pertence. O trabalho conseguiu uma acurácia de $98,43\%$, 4,21 pp acima da nossa acurácia. A utilização da CNN como extrator de características e a utilização de uma LSTM para classificação se mostrou uma combinação muito eficiente quando comparado com 3 camadas FC para classificação.

Quando comparamos os resultados, verificamos que a arquitetura proposta desempenha o reconhecimento da LSA64 de maneira satisfatória, notando ainda que o pior resultado alcançado ainda supera vários dos trabalhos relacionados. O único trabalho relacionado que obteve resultados superiores, sem retirar indivíduos da base de vídeos, utilizou uma base diferente da utilizada neste trabalho.

6 Conclusão

O reconhecimento de línguas de sinais por tradutores humanos ainda é uma solução onerosa e que necessita de muita experiência por parte do profissional. Reconhecedores automáticos destes códigos de linguagem são uma solução viável para este problema. Essas soluções podem auxiliar as pessoas sem o domínio de uma LS no aprendizado da mesma, bem como permitir sua comunicação com deficientes auditivos.

O objetivo deste trabalho foi apresentar uma metodologia computacional para o reconhecimento de gestos da LSA através de uma técnica de *deep learning*. A metodologia se utilizou de uma arquitetura 3D CNN que foi treinada e testada utilizando a base LSA64, que contém 64 classes de sinais da LSA.

Para avaliar a melhor arquitetura 3D CNN foram realizados vários testes, do número de camadas de convolução, quantidade de filtros por camada, bem como a quantidade de camadas de classificação. Após definir a melhor configuração para a rede, realizamos 10 experimentos dividindo da base de vídeos em 80% para treino e 20% para teste.

Um número considerável de classes obteve 100% de sensibilidade ou precisão, correspondendo a $15,6\%$ e $18,7\%$ das 64 classes de sinais, respectivamente. Por outro lado, a rede se mostrou incapaz de definir bem as classes que apresentavam padrões de movimento muito semelhantes, como foi o caso das duplas Vermelho-Verde e Comida-Água.

Todavia, alguns fatores limitam a extensão dos resultados aqui apresentados, como a utilização de uma base de vídeos formada por indivíduos sem o domínio da LSA, bem como o fato da validação da metodologia ser feita em uma única base de vídeos, somando-se a isto a estimação manual da topologia da rede e dos hiperparâmetros.

Levando em consideração os trabalhos relacionados, há um grande indício que a metodologia proposta é promissora no reconhecimento de língua sinais, superando a maioria dos comparativos, apresentando resultados promissores.

6.1 Trabalhos Futuros

Visando solucionar estas limitações e dado o grande interesse da comunidade científica, sugerimos como trabalhos futuros:

1. Aplicar um pré-processamento que remova informações irrelevantes, como o fundo dos vídeos;

2. Utilizar a metodologia com outras bases de vídeos, incluindo as bases utilizadas por

trabalhos relacionados, a fim de valida-la com diferentes tipos de entrada, bem como com diferentes níveis de domínio dos sinalizadores;

3. Utilizar mais vídeos por indivíduo, utilizando cada canal RGB individualmente ou até adicionar vídeos de profundidade;

4. Combinar o poder da extração de características temporais da 3D CNN com classificadores mais robustos como a SVM;

5. Utilizar duas redes 3D CNN, cada uma responsável por extrair características diferentes de uma mesma entrada, posteriormente combinando o vetor de características resultante e utilizando-o em um classificador;

6. Criar uma base de vídeos para a Libras, visando estudos mais específicos no Brasil.

7. Utilizar técnicas de meta aprendizagem para estimação da topologia e hiperparâmetros da rede.

Referências

ABREU, J. G.; TEIXEIRA, J. M.; FIGUEIREDO, L. S.; TEICHRIEB, V. Evaluating sign language recognition using the myo armband. In: IEEE. *Virtual and Augmented Reality (SVR), 2016 XVIII Symposium on*. Gramado, Brazil, 2016. p. 64–70.

ALMEIDA, S. G. M.; GUIMARÃES, F. G.; RAMÍREZ, J. A. Feature extraction in brazilian sign language recognition based on phonological structure and using rgb-d sensors. *Expert Systems with Applications*, Elsevier, v. 41, n. 16, p. 7259–7271, 2014.

ANJO, M. d. S.; PIZZOLATO, E. B.; FEUERSTACK, S. A real-time system to recognize static gestures of brazilian sign language (libras) alphabet using kinect. In: BRAZILIAN COMPUTER SOCIETY. *Proceedings of the 11th Brazilian Symposium on Human Factors in Computing Systems*. Cuiaba, Brazil, 2012. p. 259–268.

BARROS, P.; MACIEL-JUNIOR, N. T.; FERNANDES, B. J.; BEZERRA, B. L.; FERNANDES, S. M. A dynamic gesture recognition and prediction system using the convexity approach. *Computer Vision and Image Understanding*, Elsevier, v. 155, p. 139–149, 2017.

BASTOS, I. L.; ANGELO, M. F.; LOULA, A. C. Recognition of static gestures applied to brazilian sign language (libras). In: IEEE. *Graphics, Patterns and Images (SIBGRAPI), 2015 28th SIBGRAPI Conference on*. Salvador, Brazil, 2015. p. 305–312.

BEDREGAL, B. R. C.; DIMURO, G. P. et al. Interval fuzzy rule-based hand gesture recognition. In: IEEE. *Scientific Computing, Computer Arithmetic and Validated Numerics, 2006. SCAN 2006. 12th GAMM-IMACS International Symposium on*. Duisburg, Germany, 2006. p. 12–12.

BELGACEM, S.; CHATELAIN, C.; PAQUET, T. Gesture sequence recognition with one shot learned crf/hmm hybrid model. *Image and Vision Computing*, Elsevier, v. 61, p. 12–21, 2017.

BRAGATTO, T.; RUAS, G.; LAMAR, M. Real-time video based finger spelling recognition system using low computational complexity artificial neural networks. In: IEEE. *Telecommunications Symposium, 2006 International*. Fortaleza, Ceara, Brazil, 2006. p. 393–397.

BRASHEAR, H.; HENDERSON, V.; PARK, K.-H.; HAMILTON, H.; LEE, S.; STARNER, T. American sign language recognition in game development for deaf children. In: ACM. *Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility*. Portland, Oregon, USA, 2006. p. 79–86.

CARNEIRO, S. B.; SANTOS, E. D. d. M.; TALLES, M. d. A.; FERREIRA, J. O.; ALCALÁ, S. G. S.; ROCHA, A. F. D. Static gestures recognition for brazilian sign language with kinect sensor. In: IEEE. *SENSORS, 2016 IEEE*. Orlando, FL, USA, 2016. p. 1–3.

CHANSRI, C.; SRINONCHAT, J. Hand gesture recognition for thai sign language in complex background using fusion of depth and color video. *Procedia Computer Science*, Elsevier, v. 86, p. 257–260, 2016.

CHEN, Q.; GEORGANAS, N. D.; PETRIU, E. M. Real-time vision-based hand gesture recognition using haar-like features. In: IEEE. *Instrumentation and Measurement Technology Conference Proceedings, 2007. IMTC 2007. IEEE*. Warsaw, Poland, 2007. p. 1–6.

CHOLLET, F. et al. *Keras*. [S.l.]: GitHub, 2015. <https://github.com/keras-team/keras>. Acessado em: 02/01/2018.

CIREGAN, D.; MEIER, U.; SCHMIDHUBER, J. Multi-column deep neural networks for image classification. In: IEEE. *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. Providence, Rhode Island, 2012. p. 3642–3649.

CIREŞAN, D. C.; GIUSTI, A.; GAMBARDELLA, L. M.; SCHMIDHUBER, J. Mitosis detection in breast cancer histology images with deep neural networks. In: SPRINGER. *International Conference on Medical Image Computing and Computer-assisted Intervention*. Tsukuba Science City, Japan, 2013. p. 411–418.

DAHL, G. E.; YU, D.; DENG, L.; ACERO, A. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on audio, speech, and language processing*, IEEE, v. 20, n. 1, p. 30–42, 2012.

DALTON, D. S.; CRUICKSHANKS, K. J.; KLEIN, B. E.; KLEIN, R.; WILEY, T. L.; NONDAHL, D. M. The impact of hearing loss on quality of life in older adults. *The gerontologist*, Oxford University Press, v. 43, n. 5, p. 661–668, 2003.

DIAS, D. B.; MADEO, R. C.; ROCHA, T.; BÍSCARO, H. H.; PERES, S. M. Hand movement recognition for brazilian sign language: a study using distance-based neural networks. In: IEEE. *Neural Networks, 2009. IJCNN 2009. International Joint Conference on*. Atlanta, GA, USA, 2009. p. 697–704.

DIAS, J. M. S.; NANDE, P.; BARATA, N.; CORREIA, A. Ogre-open gestures recognition engine. In: IEEE. *Computer graphics and image processing, 2004. Proceedings. 17th Brazilian Symposium on*. Curitiba, Brazil, Brazil, 2004. p. 33–40.

ESCOBEDO-CARDENAS, E.; CAMARA-CHAVEZ, G. A robust gesture recognition using hand local data and skeleton trajectory. In: IEEE. *Image Processing (ICIP), 2015 IEEE International Conference on*. Quebec City, QC, Canada, 2015. p. 1240–1244.

ESTREBOU, C.; LANZARINI, L.; HASPERUÉ, W. Voice recognition based on probabilistic som. In: UNA. *Proceedings of the Conference: XXXVI Conferencia Latinoamericana en Informática, At Asunción, Paraguay*. Asunción, Paraguay, 2010.

GLOROT, X.; BORDES, A.; BENGIO, Y. Deep sparse rectifier neural networks. In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. Fort Lauderdale, FL, USA: PMLR, 2011. p. 315–323.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. MIT Press, 2016. Disponível em: <http://www.deeplearningbook.org/>.

GRAVES, A.; MOHAMED, A.-r.; HINTON, G. Speech recognition with deep recurrent neural networks. In: IEEE. *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*. Vancouver, BC, Canada, 2013. p. 6645–6649.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. In: IEEE. *Proceedings of the IEEE conference on computer vision and pattern recognition*. Las Vegas, NV, USA, 2016. p. 770–778.

HEAP, T.; HOGG, D. Towards 3d hand tracking using a deformable model. In: IEEE. *Automatic Face and Gesture Recognition, 1996., Proceedings of the Second International Conference on*. Killington, VT, USA, USA, 1996. p. 140–145.

HERNANDEZ-REBOLLAR, J. L. Gesture-driven american sign language phraselator. In: ACM. *Proceedings of the 7th international conference on Multimodal interfaces*. Torento, Italy, 2005. p. 288–292.

HINTON, G.; DENG, L.; YU, D.; DAHL, G. E.; MOHAMED, A.-r.; JAITLY, N.; SENIOR, A.; VANHOUCKE, V.; NGUYEN, P.; SAINATH, T. N. et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, IEEE, v. 29, n. 6, p. 82–97, 2012.

HINTON, G. E.; SALAKHUTDINOV, R. R. Reducing the dimensionality of data with neural networks. *science*, American Association for the Advancement of Science, v. 313, n. 5786, p. 504–507, 2006.

HUANG, F.; SUN, Z.; XU, Q.; SZE, F. Y. B.; LAN, T. W.; WANG, X. Real-time sign language recognition using rgbd stream: spatial-temporal feature exploration. In: ACM. *Proceedings of the 2nd ACM symposium on Spatial user interaction*. Honolulu, Hawaii, USA, 2014. p. 149–149.

HUANG, J.; ZHOU, W.; LI, H.; LI, W. Sign language recognition using 3d convolutional neural networks. In: IEEE. *Multimedia and Expo (ICME), 2015 IEEE International Conference on*. Turin, Italy, 2015. p. 1–6.

HUANG, R. Some inequalities for the hadamard product and the fan product of matrices. *Linear Algebra and its applications*, Elsevier, v. 428, n. 7, p. 1551–1559, 2008.

JANGYODSUK, P.; CONLY, C.; ATHITSOS, V. Sign language recognition using dynamic time warping and hand shape distance based on histogram of oriented gradient features. In: ACM. *Proceedings of the 7th International Conference on PErvasive Technologies Related to Assistive Environments*. Rhodes, Greece, 2014. p. 50.

JI, S.; XU, W.; YANG, M.; YU, K. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 35, n. 1, p. 221–231, 2013.

JIANG, F.; YAO, H.; YAO, G. Multilayer architecture in sign language recognition system. In: ACM. *Proceedings of the 6th international conference on Multimodal interfaces*. State College, PA, USA, 2004. p. 352–353.

KARPATHY, A.; FEI-FEI, L. Deep visual-semantic alignments for generating image descriptions. In: IEEE. *Proceedings of the IEEE conference on computer vision and pattern recognition*. Honolulu, Hawaii, 2015. p. 3128–3137.

KUMAR, P.; GAUBA, H.; ROY, P. P.; DOGRA, D. P. Coupled hmm-based multi-sensor data fusion for sign language recognition. *Pattern Recognition Letters*, Elsevier, v. 86, p. 1–8, 2016.

KUMAR, P.; GAUBA, H.; ROY, P. P.; DOGRA, D. P. A multimodal framework for sensor based sign language recognition. *Neurocomputing*, Elsevier, 2017.

LASKAR, M. A.; DAS, A. J.; TALUKDAR, A. K.; SARMA, K. K. Stereo vision-based hand gesture recognition under 3d environment. *Procedia Computer Science*, Elsevier, v. 58, p. 194–201, 2015.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *nature*, Nature Publishing Group, v. 521, n. 7553, p. 436, 2015.

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, IEEE, v. 86, n. 11, p. 2278–2324, 1998.

LI, K.; ZHOU, Z.; LEE, C.-H. Sign transition modeling and a scalable solution to continuous sign language recognition for real-world applications. *ACM Transactions on Accessible Computing (TACCESS)*, ACM, v. 8, n. 2, p. 7, 2016.

LIANG, W.; GUIXI, L.; HONGYAN, D. Dynamic and combined gestures recognition based on multi-feature fusion in a complex environment. *The Journal of China Universities of Posts and Telecommunications*, Elsevier, v. 22, n. 2, p. 81–88, 2015.

LIM, K. M.; TAN, A. W.; TAN, S. C. Block-based histogram of optical flow for isolated sign language recognition. *Journal of Visual Communication and Image Representation*, Elsevier, v. 40, p. 538–545, 2016.

LIM, K. M.; TAN, A. W.; TAN, S. C. A feature covariance matrix with serial particle filter for isolated sign language recognition. *Expert Systems with Applications*, Elsevier, v. 54, p. 208–218, 2016.

LIMA, M.; NETO, P.; VIDAL, R.; LIMA, G.; SANTOS, J. Libras translator via web for mobile devices. In: ACM. *Proceedings of the 6th Euro American Conference on Telematics and Information Systems*. Valencia, Spain, 2012. p. 399–402.

LIU, J.; ZHONG, L.; WICKRAMASURIYA, J.; VASUDEVAN, V. uwave: Accelerometer-based personalized gesture recognition and its applications. *Pervasive and Mobile Computing*, Elsevier, v. 5, n. 6, p. 657–675, 2009.

MADEO, R. C.; PERES, S. M.; DIAS, D. B.; BOSCARIOLI, C. Gesture recognition for fingerspelling applications: an approach based on sign language cheremes. In: ACM. *Proceedings of the 12th international ACM SIGACCESS conference on Computers and accessibility*. Orlando, Florida, USA, 2010. p. 261–262.

MADEO, R. C.; PERES, S. M.; LIMA, C. A.; BOSCARIOLI, C. Hybrid architecture for gesture recognition: Integrating fuzzy-connectionist and heuristic classifiers using fuzzy syntactical strategy. In: IEEE. *Neural Networks (IJCNN), The 2012 International Joint Conference on*. Brisbane, QLD, Australia, 2012. p. 1–8.

MARIN, G.; DOMINIO, F.; ZANUTTIGH, P. Hand gesture recognition with leap motion and kinect devices. In: IEEE. *Image Processing (ICIP), 2014 IEEE International Conference on*. Paris, France, 2014. p. 1565–1569.

MASOOD, S.; SRIVASTAVA, A.; THUWAL, H. C.; AHMAD, M. Real-time sign language gesture (word) recognition from video sequences using cnn and rnn. In: *Intelligent Engineering Informatics*. Singapore: Springer, 2018. p. 623–632.

MITRA, S.; ACHARYA, T. Gesture recognition: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, IEEE, v. 37, n. 3, p. 311–324, 2007.

MNIH, V.; BADIA, A. P.; MIRZA, M.; GRAVES, A.; LILLICRAP, T.; HARLEY, T.; SILVER, D.; KAVUKCUOGLU, K. Asynchronous methods for deep reinforcement learning. In: *International Conference on Machine Learning*. New York, New York, USA: PMLR, 2016. p. 1928–1937.

MOELLER, M. P. Early intervention and language development in children who are deaf and hard of hearing. *Pediatrics*, Am Acad Pediatrics, v. 106, n. 3, p. e43–e43, 2000.

MOLCHANOV, P.; GUPTA, S.; KIM, K.; KAUTZ, J. Hand gesture recognition with 3d convolutional neural networks. In: IEEE. *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. Boston, MA, USA, 2015. p. 1–7.

MOLCHANOV, P.; GUPTA, S.; KIM, K.; PULLI, K. Multi-sensor system for driver's hand-gesture recognition. In: IEEE. *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*. Ljubljana, Slovenia, 2015. v. 1, p. 1–8.

NEIVA, D. H.; ZANCHETTIN, C. A dynamic gesture recognition system to translate between sign languages in complex backgrounds. In: IEEE. *Intelligent Systems (BRACIS), 2016 5th Brazilian Conference on*. Recife, Brazil, 2016. p. 421–426.

NETO, F. M. de P.; CAMBUIM, L. F.; MACIEIRA, R. M.; LUDERMIR, T. B.; ZANCHETTIN, C.; BARROS, E. N. Extreme learning machine for real time recognition of brazilian sign language. In: IEEE. *Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on*. Kowloon, China, 2015. p. 1464–1469.

OHN-BAR, E.; TRIVEDI, M. M. Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations. *IEEE transactions on intelligent transportation systems*, IEEE, v. 15, n. 6, p. 2368–2377, 2014.

OYEDOTUN, O. K.; KHASHMAN, A. Deep learning in vision-based static hand gesture recognition. *Neural Computing and Applications*, Springer, v. 28, n. 12, p. 3941–3951, 2017.

PIGOU, L.; DIELEMAN, S.; KINDERMANS, P.-J.; SCHRAUWEN, B. Sign language recognition using convolutional neural networks. In: SPRINGER. *Workshop at the European Conference on Computer Vision*. Zurich, Switzerland, 2014. p. 572–578.

POLYAK, B. T. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, Elsevier, v. 4, n. 5, p. 1–17, 1964.

PORFIRIO, A. J.; WIGGERS, K. L.; OLIVEIRA, L. E.; WEINGAERTNER, D. Libras sign language hand configuration recognition based on 3d meshes. In: IEEE. *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*. Manchester, UK, 2013. p. 1588–1593.

RIEDMILLER, M.; BRAUN, H. A direct adaptive method for faster backpropagation learning: The rprop algorithm. In: IEEE. *Neural Networks, 1993., IEEE International Conference on*. San Francisco, CA, USA, USA, 1993. p. 586–591.

RONCHETTI, F. *Reconocimiento de gestos dinámicos y su aplicación al lenguaje de señas*. Tese (Doutorado) — Facultad de Informática, 2017.

RONCHETTI, F.; QUIROGA, F.; ESTREBOU, C.; LANZARINI, L.; ROSETE, A. Lsa64: A dataset of argentinian sign language. *XX II Congreso Argentino de Ciencias de la Computación (CACIC)*, 2016.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

SPIEGEL, M. R.; SCHILLER, J. J.; SRINIVASAN, R. A. *Probabilidade e Estatística-: Coleção Schaum*. New York, New York, USA: Bookman Editora, 2016.

STARNER, T.; WEAVER, J.; PENTLAND, A. Real-time american sign language recognition using desk and wearable computer based video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE, v. 20, n. 12, p. 1371–1375, 1998.

THALANGE, A.; DIXIT, S. Cohst and wavelet features based static asl numbers recognition. *Procedia Computer Science*, Elsevier, v. 92, p. 455–460, 2016.

Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, maio 2016. Disponível em: <http://arxiv.org/abs/1605.02688>.

TIELEMAN, T.; HINTON, G. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, v. 4, n. 2, p. 26–31, 2012.

WANG, H.; WANG, N.; YEUNG, D.-Y. Collaborative deep learning for recommender systems. In: ACM. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Sydney, NSW, Australia, 2015. p. 1235–1244.

WANG, M.; CHEN, W.-Y.; LI, X. D. Hand gesture recognition using valley circle feature and hu's moments technique for robot movement control. *Measurement*, Elsevier, v. 94, p. 734–744, 2016.

WANG, X.; WANG, Y. Improving content-based and hybrid music recommendation using deep learning. In: ACM. *Proceedings of the 22nd ACM international conference on Multimedia*. Orlando, Florida, USA, 2014. p. 627–636.

WU, Y.; HUANG, T. S. Vision-based gesture recognition: A review. In: SPRINGER. *International Gesture Workshop*. Gif-sur-Yvette, France, 1999. p. 103–115.

XU, R.; ZHOU, S.; LI, W. J. Mems accelerometer based nonspecific-user hand gesture recognition. *IEEE sensors journal*, IEEE, v. 12, n. 5, p. 1166–1173, 2012.

YANG, S.; ZHU, Q. Continuous chinese sign language recognition with cnn-lstm. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. *Ninth International Conference on Digital Image Processing (ICDIP 2017)*. Hong Kong, China, 2017. v. 10420, p. 104200F.