Universidade Federal do Pampa
Jorge Daniel Barros Junior

Tradução Automática de Línguas de Sinais: do Sinal para a Escrita

Alegrete
2016

Jorge Daniel Barros Junior

Tradução Automática de Línguas de Sinais: do Sinal para a Escrita

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação da Universidade Federal do Pampa como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Marcelo Resende Thielo

Coorientador: Prof. Dr. Fábio Natanael Kepler

Alegrete
2016

Jorge Daniel Barros Junior

Tradução Automática de Línguas de Sinais: do Sinal para a Escrita

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação da Universidade Federal do Pampa como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Trabalho de Conclusão de Curso defendido e aprovado em 22 de JUNHO de 2016

Banca examinadora:

Prof. Dr. Marcelo Resende Thielo
Orientador

Prof. Dr. Fábio Natanael Kepler
Co-orientador
UNIPAMPA

Profª Ma. Alice Fonseca Finger
UNIPAMPA

Profª Ana Paula Gomes Lara
UNIPAMPA

Resumo

O reconhecimento automático de sinais de uma Língua de Sinais (LS) por um computador ainda é um grande desafio. Até o momento, não existia uma conversão automática de sinais para a Escrita de Línguas de Sinais (ELS) na notação em SignWriting (SW). Esses fatores foram os que motivaram o desenvolvimento deste trabalho a fim de encontrar uma solução para esse problema que ainda não havia sido solucionado na literatura. Dessa forma, neste trabalho fizemos o reconhecimento de sinais em imagens de uma LS e convertemos para a escrita de sinais em SW. Mais precisamente, nos concentramos no parâmetro de configuração de mão, que é importante para definir o formato que a mão irá assumir durante a execução de um sinal. Para isso, utilizamos Redes Neurais Artificiais do tipo Convolutional Neural Network (CNN) que é um modelo de aprendizado de máquina que nos possibilitou fazer o reconhecimento dos sinais. Para fazer a captura das imagens, utilizamos o sensor de movimentos Kinect em conjunto com uma aplicação desenvolvida que faz o preprocessamento das imagens. Fizemos o treinamento da CNN com 16 configurações de mão diferentes e obtivemos uma acurácia satisfatória de 87.5%.

Palavras-chave: Escrita de Sinais. SignWriting. Aprendizado de Máquina. Rede Neural Convolucional. Kinect.

Abstract

Automatic recognition of signs of a Sign Language by a computer is still a challenge. So far, there is no automatic conversion signals for sign language written in the notation in SignWriting (SW). These factors were that motivated the development of this work in order to find a solution to this problem that had not been solved in the literature. Thus, in this work we recognized language signals in images and converted them into written signals known as SW. More precisely, we focused on the parameter of hand shape, which is important to define the hand shape that the hand will assume during the execution of a signal. For this, we trained Convolutional Neural Network (CNN) that is a mechanism of Machine Learning (ML) which provided the signal recognition. To capture images, we used the Kinect motion sensor together with an developed application that does the preprocessing of the images. We did training the CNN with 16 different hand shapes and have achieved the average accuracy of 87.5%.

Key-words: Write Signals. SignWriting. Machine Learning. Convolutional Neural Network. Kinect.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas

ASL American Sign Language

CNN Convolutional Neural Network

CSL Chinese Sign Language

DAC Deaf Action Movement Writing

DL Deep Learning

EliS Sistema de Escrita das Línguas de Sinais

ELS Escrita de Línguas de Sinais

FSW Formal SignWriting

HamNoSys Hamburg Notation System

INES Instituto Nacional de Educação de Surdos

ISL Indonesian Sign Language

LIBRAS Língua Brasileira de Sinais

LS Língua de Sinais

LSF Língua de Sinais Francesa

ML Machine Learning

MLP Perceptron Multicamadas

NFN Notação de François Neve

NS Notação de Stokoe

ReLU Retified Linear Unit

RNA Rede Neural Artificial

SDK Software Development Kit

SVM Support Vector Machine

SW SignWriting

Sumário

1 Introdução

Existem comunidades formadas por surdos, também chamadas de *comunidades surdas*, compostas de milhões de pessoas ao redor do mundo. Aproximadamente 360 milhões de pessoas no mundo possuem algum tipo de problema de audição e uma parcela dessas pessoas são totalmente surdas (ORGANIZATION, 2015). Devido a necessidade das pessoas se comunicarem, diversas línguas foram sendo criadas ao longo dos anos. As línguas podem se manifestar de maneira oral ou gestual. A língua oral, também conhecida por oral-auditiva, é transmitida através da fala e recebida através da audição. A língua gestual, também conhecida por espaço-visual, é transmitida através de sinais e recebida através da visão.

A comunidade surda utiliza como meio de comunicação a língua espaço-visual, mais especificamente a Língua de Sinais (LS). Existe uma diversidade grande de LS, onde cada uma possui características próprias. No Brasil, a Língua Brasileira de Sinais (LIBRAS) é a LS oficial (CIVIL, 2002).

No caso da escrita, os surdos foram se adaptando a aprender a escrita das línguas orais. Entretanto, a escrita das línguas orais seguem regras fonológicas que os surdos possuem dificuldades de compreender, isso porque a comunicação deles não segue essas mesmas regras. Outro fator é que LS não era vista como uma língua onde fosse possível escrevê-la, já que é constituída de sinais visuais.

Com o passar dos anos, foram criadas escritas baseadas nas LSs. Essas escritas exploravam os parâmetros utilizados na LS para expressá-los de forma completa. O sistema de escrita de sinais chamado SignWriting (SW) é o que mais se destacou e será o nosso foco neste trabalho. O SW é universal, ou seja, sua estrutura permite representar qualquer LS. Os símbolos do SW são facilmente representados no computador utilizando uma linguagem regular chamada Formal SignWriting (FSW).

Ainda não existe uma conversão automática da LS para a escrita em SW. Logo, a única maneira de escrever em SW é utilizando algumas ferramentas manuais disponíveis na internet, como o *SignMaker*[*]. Porém, essas ferramentas não são muito práticas e requerem bastante esforço para escrever os sinais em SW.

Alguns trabalhos, que serão descritos no Capítulo 3, propuseram fazer o reconhecimento de sinais para a escrita em línguas orais. Utilizaram o sensor de movimentos Kinect para capturar os sinais e processaram as imagens obtidas aplicando mecanismos de aprendizado de máquina.

[*]Disponível em: http://www.signbank.org/signmaker/

O principal objetivo deste trabalho é fazer o reconhecimento de sinais em imagens de uma LS para convertê-los na escrita de sinais em SW. Entretanto, nos concentramos em reconhecer o parâmetro de configuração de mão.

Neste trabalho, no Capítulo 2 descrevemos a base teórica sobre LS, escrita de sinais e aprendizado de máquina. No Capítulo 3 apontamos alguns trabalhos relacionados ao nosso. No Capítulo 4 apresentamos a nossa metodologia desenvolvida no trabalho juntamente com os resultados obtidos, explicando o processo para o reconhecimento dos sinais em imagens e a conversão dos mesmos em SW. Por fim, no Capítulo 5 discutimos nossas conclusões sobre o trabalho desenvolvido e descrevemos os trabalhos futuros.

2 Fundamentação

Este Capítulo é destinado a descrever alguns conceitos básicos e fundamentais para o entendimento do nosso trabalho. A seção 2.1 irá abordar a parte conceitual sobre Língua de Sinais (LS), a seção 2.2 irá introduzir os conceitos relacionados à Escrita de Línguas de Sinais (ELS) e na seção 2.3 explicaremos os fundamentos relativos ao Aprendizado de Máquina com ênfase nas Redes Neurais Artificiais (RNAs).

2.1 Língua de Sinais

A língua é um meio de comunicação composto por palavras, sinais e expressões que seguem determinadas regras. A manifestação da língua pode ser dada de maneira oral ou gestual. A Língua de Sinais (LS) é uma língua constituída por sinais. Diferentemente da língua oral-auditiva, a língua de sinais é considerada espaço-visual, sendo transmitida através de expressões corporais e compreendida através da visão (PEREIRA, 2010).

Segundo Dizeu e Caporali (2005), a LS é considerada uma língua natural para os surdos, já que é adquirida espontaneamente, sem a necessidade de um treinamento. A LS durante um longo período não foi considerada como uma língua, pois acreditavam que seria apenas uma maneira de expressar gestualmente as palavras de uma língua oral-auditiva, com o propósito de substituir a fala. Além disso, diziam que a LS não seguia nenhum tipo de regra específica.

Entretanto, a LS possui gramática, semântica, pragmática, entre outras características que comprovam que a LS é de fato uma língua como qualquer língua oral-auditiva (PEREIRA, 2010). Além disso, por volta de 1960 através de alguns estudos que comparavam a gramática da American Sign Language (ASL) com a língua inglesa foi visto que a LS possuía uma gramática própria (PERLMUTTER, 2013). Um simples teste foi feito onde era escolhida uma palavra em inglês que possuía mais de um significado como no caso da palavra “right” que em inglês poderia ser entendido por “correto” ou “direita”. Porém, em ASL não existe um sinal único que simbolize esses dois significados. Portanto, a LS expressa significados e não palavras em línguas orais-auditivas (PERLMUTTER, 2013).

Assim como na língua oral, a LS possui variações, e dependendo do país, região e até mesmo comunidade, a LS pode variar bastante. Por exemplo, nos Estados Unidos a LS é a ASL, na França é Língua de Sinais Francesa (LSF) e no Brasil é a Língua Brasileira de Sinais (LIBRAS) que será mais detalhada na próxima subseção.

2.1.1 Língua Brasileira de Sinais (LIBRAS)

Segundo Honora e Frizanco (2009), a LIBRAS teve origem da LSF, durante o segundo império, devido ao fato de que o filho francês da Princesa Isabel chamado Hernest Huet era surdo. Com isso, Hernest Huet trouxe da França o alfabeto manual francês juntamente com a LSF. Logo após, em 1857 houve a fundação no Rio de Janeiro do Instituto Nacional de Educação de Surdos (INES) (SANTOS; GODOI; SILVA, 2013). De acordo com Santos, Godoi e Silva (2013), em 1880, houve a proibição pelo Congresso de Milão de qualquer língua gestual para a alfabetização nas escolas.

Atualmente, a LIBRAS é a LS utilizada no Brasil pelos surdos brasileiros. Porém, só foi reconhecida oficialmente como uma língua no Brasil em 2002, com a Lei 10.436/2002 (CIVIL, 2002). A LIBRAS possui variações dentro do Brasil assim como no português que em estados distintos possuem gírias locais. Em LIBRAS não é diferente, e é comum existirem diferentes sinais que representem o mesmo significado.

Para formar um sinal em LIBRAS é necessário combinar parâmetros. Os parâmetros são divididos em cinco categorias:

1. **Configuração de Mão:** são formatos que a mão pode executar, podendo ser executado com a mão esquerda, direita ou ambas. Conforme ilustra a Figura 1, existem diversos formatos de mão possíveis, dentre eles 26 são para o alfabeto, utilizado para soletrações (datilologia).

Figura 1: Configurações de Mãos em LIBRAS (INES, 2011).

2. **Ponto de Articulação:** é o parâmetro que indica onde a mão irá tocar em alguma parte do corpo ou ficar em um espaço neutro.

3. **Movimento:** é a locomoção da mão no espaço. Os sinais podem possuir movimentação ou não. Podemos classificar os movimentos como: retilíneo, helicoidal, circular, semicircular, sinuoso e angular (PEREIRA, 2010).

4. **Orientação da Palma:** é indicada pela direção e sentido com que a palma da mão está. Em alguns sinais pode-se variar a direção da palma da mão durante o movimento. Além disso, o sentido oposto de um sinal pode simbolizar o significado contrário de um sinal, como ilustrado na Figura 2, onde os verbos IR, VIR e SUBIR, DESCER são opostos de direção e sentido.

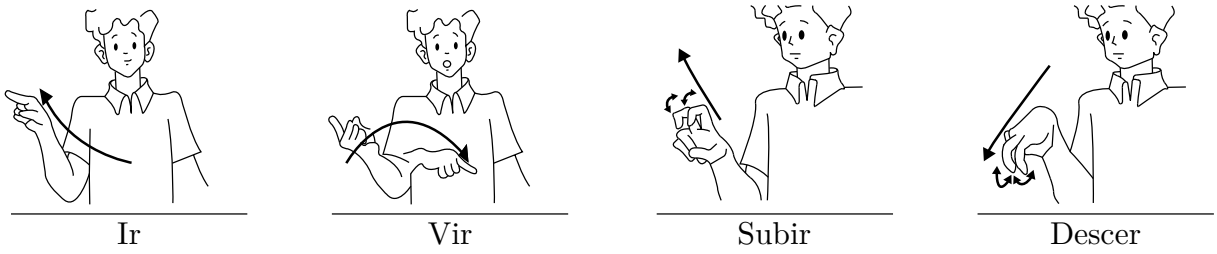


Figura 2: Exemplos de sinais com sentido e direção oposta. Adaptado de Cossio et al. (2012).

5. **Expressões Não-Manuais:** são compostas de expressão faciais e corporais e a transmissão é feita através do rosto, olho, cabeça ou tronco. Possuem o propósito de dar entonação nos sinais e determinar o significado do sinal.

2.2 Escrita de Sinais

Para crianças ouvintes é intuitivo aprender a escrita das línguas orais devido as propriedades fonológicas contidas nesse tipo de língua (BARRETO; BARRETO, 2012). Dessa maneira, a criança ouvinte e a escrita se complementam, fazendo com que o aprendizado se torne mais completo. Por exemplo, quando uma criança ouve uma palavra, logo ela consegue escrever como se fala. Esse processo facilita aprendizagem da escrita para as crianças ouvintes.

Entretanto, quando pensamos em crianças surdas, esse processo não passa a ser mais um processo natural de aprendizagem. Isso porque, a criança surda utiliza a LS que é uma língua espaço-visual, a qual não se aproveita das propriedades fonológicas que uma língua oral possui. Com isso, a alfabetização dessas crianças surdas é prejudicada.

A Escrita de Línguas de Sinais (ELS) é uma maneira de se escrever os sinais. Diferentemente das línguas orais, onde a escrita alfabética é um padrão de escrita. Na LS, não existe uma maneira padronizada de ELS. Portanto, existem diversas técnicas de se representar uma ELS. Algumas das principais técnicas são:

- **Notação Mimographie:** criada pelo francês Auguste Bébian em 1822 que afirmava que era fundamental uma ELS para que pudesse ser utilizada como ferramenta de ensino (BARRETO; BARRETO, 2012). Bébian identificou a necessidade de se dividir em cinco elementos básicos: configuração da mão, posição no espaço, local

de execução do sinal, ação executada e a expressão facial (OVIEDO, 2007). A Tabela 1 mostra os movimentos que para Bébian poderiam ser de seis tipos.

Movimento	**Símbolo**
Da esquerda para direita	⊂
Da direita para esquerda	⊃
De baixo para cima	∪
De cima para baixo	∩
Para frente	⊖
Para trás	⦶

Tabela 1: Tipos de movimentos em Mimographie. Adaptado de Oviedo (2007).

- **Notação de Stokoe (NS):** criada pelo norte americano William Stokoe, foi o primeiro sistema de escrita para sinais do ASL e conseguiu provar que ASL era uma língua por si só (PANSE; GROMISCH, 2012). Em 1964, Stokoe lançou um dicionário com detalhes de sua notação. A NS original possui 55 símbolos, divididos nos parâmetros localização, movimentação e configuração de mão, que posteriormente foi visto que não eram suficientes (MARTIN, 2000). A NS foi base para outras notações, sendo modificada conforme as necessidades. Alguns caracteres da NS são mostrados na Figura 3.

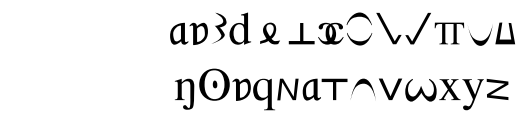


Figura 3: Alguns caracteres da NS.

- **Hamburg Notation System (HamNoSys):** criada na Universidade de Hamburgo (Alemanha) e baseada na NS, a HamNoSys é um sistema alfabético que descreve sinais em um nível fonético (HANKE, 2004). A Figura 4 mostra algumas configurações de mãos. De acordo com Hanke (2004), a HamNoSys foi projetada seguindo alguns princípios:

  - Uso internacional: HamNoSys não se limita apenas a algumas LS, logo é possível representar qualquer LS;
  - Iconicidade: podem ser criados novos glifos para facilitar a memorização ou dedução de um símbolo;
  - Economia: os sinais descritos devem seguir o uso dos princípios das condições de simetria;

– Integração com as ferramentas computacionais padrões: a HamNoSys deve ter suporte as ferramentas que trabalhem com texto;

– Sintaxe formal: a sintaxe deve ser bem definida e seguir o princípio da composicionalidade;

– Extensibilidade: HamNoSys permite que novas descrições de sinais sejam incluídas na notação.

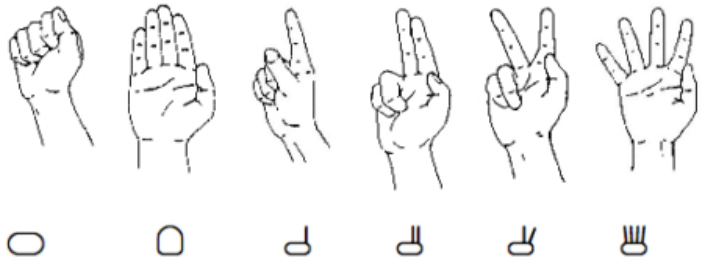


Figura 4: Configurações de Mãos na HamNoSys (HANKE, 2004).

- **Sistema D'Sign:** criado por Paul Jouison. Devido ao seu falecimento, Brigitte Garcia seguiu os estudos sobre o sistema e escreveu uma tese (STUMPF, 2005). Garcia descreve o sistema, como bem elaborado e com capacidade de transcrever completamente frases da LSF. A Figura 5 ilustra um exemplo de texto escrito no sistema D'Sign.

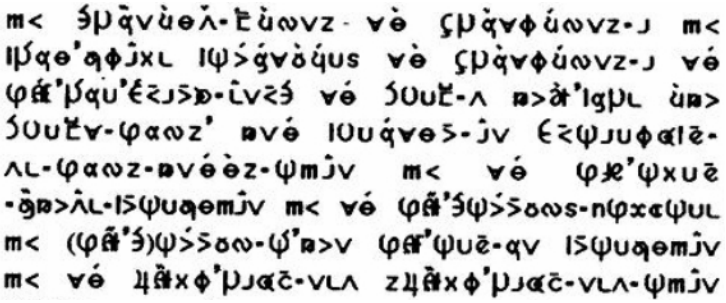


Figura 5: Texto escrito em D'Sign. Adaptado de Stumpf (2005).

- **Notação de François Neve (NFN):** Também baseada na NS, a NFN foi criada por François Xavier Neve. O padrão de escrita da NFN é verticalmente em colunas e de cima para baixo (STUMPF, 2005). Quando a escrita é feita com apenas a mão predominante é necessário apenas uma coluna, caso seja utilizada a outra mão será necessário ter duas colunas, sendo uma coluna para cada mão. A Figura 6 apresenta algumas configurações de mão em NFN.

- **Sistema de Escrita das Línguas de Sinais (EliS):** foi criado por Mariângela Estelita Barros em 1997. De acordo com Barros et al. (2008), o EliS é uma ELS

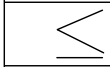
1 - 2 - 3 - 4 - 5 - 20			
A - B - C - D - E - F - G - I - L - M - N - O P - Q - R - S - T - U - V - W - X - Y - Z			Datilologia
	Bico de pardal		Asas de águia
	Cabeça de elefante		Garra de urso
	Pinça		Colher

Figura 6: Configurações de Mão em NFN. Adaptado de Stumpf (2005).

de base alfabética e linear, organizada com base nos parâmetros da NS. Barros identificou quatro parâmetros: configuração de dedos, orientação da palma, ponto de articulação e movimentação. O EliS introduziu um novo parâmetro que é a configuração de dedos, que indica a posição de cada dedo em uma configuração de mão, como ilustrado na Figura 7.

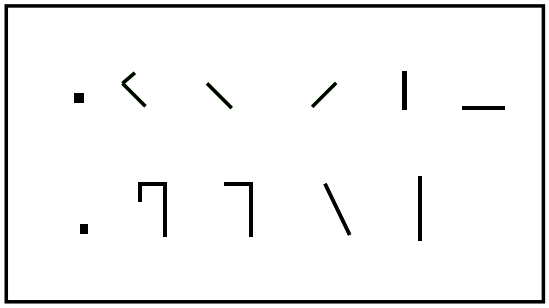


Figura 7: Configurações de Dedos no EliS (BARROS et al., 2008).

- **Sistema de escrita SignWriting (SW):** foi criado por Valerie Sutton, baseado no sistema de escrita de passos de dança chamado DanceWriting. O SW é o sistema de ELS mais utilizado no mundo, sendo utilizado em mais de 40 países (BARRETO; BARRETO, 2012). Assim como o alfabeto romano é utilizado para representar a escrita de muitas línguas orais, o SW é considerado universal porque através dele é possível representar diversas LS (GUIMARÃES; GUARDEZI; FERNANDES, 2014). A escrita é feita através de símbolos que representam configurações e movimentos das mãos, expressões faciais e deslocamentos corporais (STUMPF, 2000). A Figura 8 mostra um exemplo de escrita. Nos dias de hoje o SW está sendo desenvolvido pelo Deaf Action Movement Writing (DAC) que é uma organização dirigida por Sutton e sem fins lucrativos (BARRETO; BARRETO, 2012). O SW será mais detalhado na próxima seção.

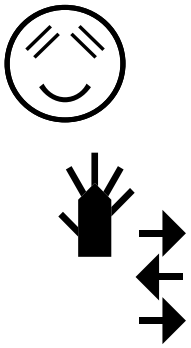


Figura 8: Escrita do “oi” em SW.

2.2.1 SignWriting

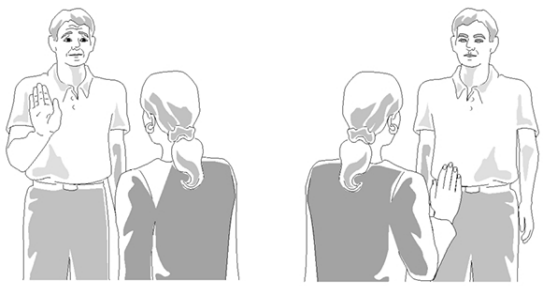
O SignWriting (SW) começou a entrar no Brasil a partir de 1996, quando foi descoberto pelo Dr. Antonio Rocha Costa que os sinais do SW eram utilizados pelo computador (QUADROS, 2005). Logo, ele organizou um grupo para estudar o SW. Apesar do crescimento do SW no Brasil, ele ainda não é considerado uma escrita oficial brasileira.

Para entender e escrever em SW é necessário estudar a sua estrutura de construção de escrita, na qual será descrita na próxima subseção.

2.2.1.1 Estrutura de escrita

A estrutura de escrita do SW é composta pelos parâmetros de orientação da palma, configuração de mão, tipos de contatos, movimento, expressões facial e localização corporal. Alguns destes parâmetros serão detalhados nesta subseção.

Inicialmente, existem duas perceptivas de visão: receptiva e expressiva, que representam (a) e (b) respectivamente na Figura 9. A visão receptiva é quando alguém está sinalizando e você está como observador, enquanto que a visão expressiva é quando você está sinalizando, logo a sua visão é a visão do sinalizador.

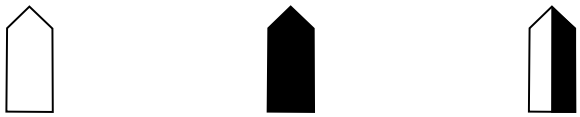


(a) (b)

Figura 9: Perspectivas de visão. Adaptado de Sutton (2009).

Na orientação da palma, a parte branca indica a palma da mão (Figura 10a) e a parte preta indica o dorso da mão (Figura 10b). Quando a mão está virada lateralmente

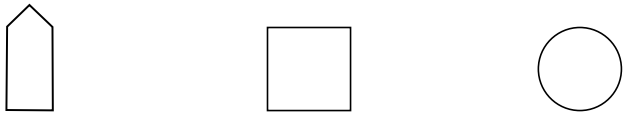
(Figura 10c), através das cores branco e preto é possível saber para onde a palma da mão está direcionada.



(a) (b) (c)

Figura 10: Orientações da Mão. Adaptado de Sutton (2009).

As configurações de mãos em SW são formadas utilizando os três símbolos básicos, como apresentados na Figura 11. Para representar a mão espalmada onde os dedos estão tocando uns aos outros é utilizado o símbolo da Figura 11a. Quando a mão está fechada, ou seja, os dedos estão tocando a palma da mão, então a representação é feita como na Figura 11b. E por último, a Figura 11c representa quando a mão está aberta com os dedos tocando entre si.



(a) (b) (c)

Figura 11: Configurações básicas de mão. Adaptado de Sutton (2009).

Existem alguns símbolos que identificam o tipo de contato durante a sinalização, a Figura 12 ilustra alguns deles. O símbolo *Tocar* é utilizado quando acontece um contato entre as mãos ou com outra parte do corpo. O símbolo *Pegar* é utilizado para descrever o ato de pegar e segurar algo. O símbolo *Contato entre* ocorre quando existe um contato entre duas partes do corpo ao mesmo tempo. O símbolo *Bater* é o mesmo conceito do símbolo *Tocar*, porém a ação é feita com mais força. O símbolo *Escovar* é utilizado quando a mão se arrasta em uma superfície e logo se afasta da mesma. O símbolo *Esfregar em círculo* define o ato de manter o contato com alguma superfície e ir fazendo um movimento circular. Existem outros símbolos que utilizam a mesma representação, porém com algumas modificações, como o símbolo "Esfregar linear", que é uma variação do símbolo *Esfregar*, porém é adicionada uma seta que indica a direção que o o movimento irá exercer.

Enquanto na maioria das línguas orais o padrão de escrita é horizontalmente, em SW a escrita é feita verticalmente. Segundo Sutton (1998), quando estamos sinalizando o nosso corpo está na posição vertical, logo faz mais sentido que a escrita em SW siga o padrão vertical. Sutton (1998) também afirma que o SW no padrão vertical torna a leitura mais fácil, aumentando a sua velocidade.

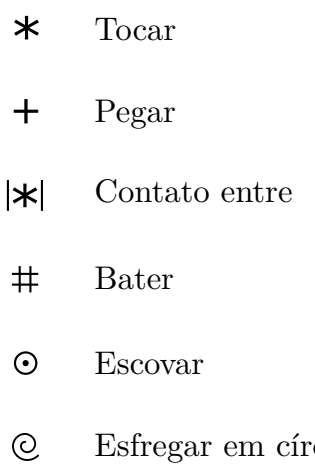


Figura 12: Tipos de contatos. Adaptado de Sutton (2009).

O padrão vertical do SW é composto por três colunas (ou pistas), como mostra a Figura 13. Os números que estão acima das colunas indicam as posições verticais entre as colunas. O número 0 indica o centro, os números 1 e 2 indicam a localização laterais para as mãos e o corpo.

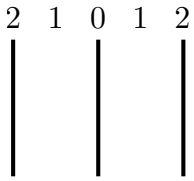


Figura 13: Escrita do SW em três colunas.

A Figura 14 mostra como é posicionado o sinal. Observe que o rosto está posicionado no centro, enquanto que as mãos estão posicionadas nas laterais mais próximas do centro. Entretanto, os sinais podem se mover entre as colunas. Na LS quando é necessário fazer comparações, os sujeitos são definidos em colunas diferentes para que haja uma comparação (SUTTON, 1998).

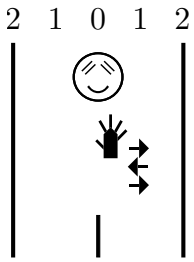


Figura 14: Exemplo de um sinal escrito nas colunas.

A estrutura técnica do SW é bastante complexa, logo existem alguns detalhes que não foram abordados nessa subseção, como os símbolos que representam movimento, pontuação, expressões faciais, entre outras características.

2.2.1.2 Formal SignWriting

O Formal SignWriting (FSW) é um conjunto de caracteres em ASCII que representam sinais logográficos. Esse conjunto de caracteres é definido através de uma expressão regular. Os sinais são escritos dentro de uma *SignBox*, que é uma área provida de coordenadas x e y no qual delimitam o espaço onde o sinal poderá aparecer (SLEVINSKI, 2015). A expressão regular utilizada para representar os sinais em FSW pode ser dividida nos seguintes conjuntos:

- **Symbol Keys:** possui o tamanho de 6 caracteres, como ilustra a Figura 15. O primeiro caractere sempre será o "S" (a). Os próximos três caracteres identificam a base do símbolo (b). Os últimos dois caracteres identificam o preenchimento (c) e a rotação (d).

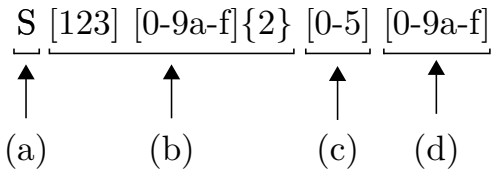
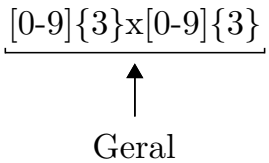
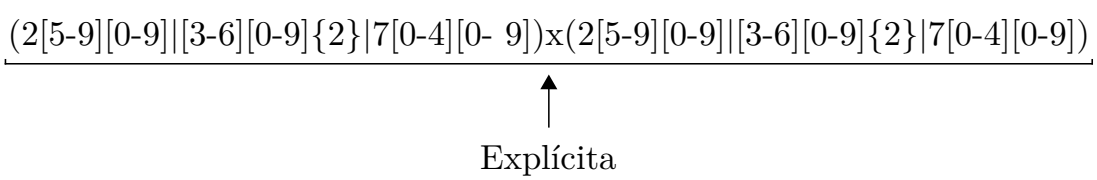


Figura 15: Expressão Regular do Símbolo.

- **Coordinates:** as coordenadas possuem dois *tokens*. O primeiro token representa a coordenada x e o segundo a coordenada y. As coordenadas podem ser de dois tipos: coordenada geral e coordenada explícita. De acordo com Slevinski (2015), a coordenada geral é mais adequada para pré-processamento. Além disso, a coordenada geral é definida por três dígitos, seguidos da letra "x", e logo após mais três dígitos. A coordenada explícita possui a mesma sequência lógica, porém, com a restrição em que os valores das coordenadas são limitadas entre um intervalo de 250 à 749.





- **SignBox:** são definidas a partir de dois parâmetros de largura e altura. Os sinais são colocados dentro da área de uma SignBox e eles podem se sobrepor. A SignBox pode ser descrita com oito tokens.

SignBox ⟶ [BLMR]([0-9]{3}x[0-9]{3})

- **Temporal Sequence:** é um prefixo opcional do SignBox, o qual representa uma lista de símbolos escritos e/ou símbolos de localização detalhada que identificam uma ordem temporal e uma análise adicional (SLEVINSKI, 2015). A expressão regular que define o temporal sequence é sempre iniciada pela letra "A", seguida de uma sequencia de um ou mais símbolos.

Temporal Sequence ⟶ (A(S[123][0-9a-f]{2}[0-5][0-9a-f])+)?

- **Sentences:** as sentenças são uma mistura de sinais e pontuações. As pontuações são um tipo de Symbol Key e servem para estruturar o texto em sentenças. É importante salientar que os símbolos de pontuação devem ser utilizadas sozinhos, sem a adição de outros sinais. Logo, para formar uma sentença é necessário utilizar simultaneamente os conceitos dos conjuntos citados anteriormente.

Pontuação ⟶ S38[7-9ab][0-5][0-9a-f][0-9]{3}x[0-9]{3}

2.3 Aprendizado de Máquina

O Aprendizado de Máquina - *Machine Learning (ML)* é uma das subáreas da Inteligência Artificial. De acordo com Mitchell (1997), ML é definido por:

> *"Um programa de computador aprende a partir da experiência E na realização de uma determinada tarefa T e com uma determinada medida de performance P, se a sua performance na realização da tarefa T, medida por P, aumenta com a experiência E."*

Outra definição de ML é descrita em Bengio e Courville (2016) como a capacidade que um computador possui de extrair padrões através de dados não tratados. Logo, a partir de um conjunto de dados, o ML através de algoritmos, busca encontrar padrões que classifiquem os dados. Com isso, podemos utilizar a capacidade computacional para resolver diversos problemas. Alguns problemas bastante conhecidos e já solucionados utilizando ML são a filtragem de spam, reconhecimento de manuscrito, entre outros.

O ML também possui subdivisões que definem a maneira na qual os algoritmos são treinados para lidar com os problemas. As principais subdivisões são Aprendizado Supervisionado e Aprendizado Não Supervisionado.

O aprendizado supervisionado é um tipo de algoritmo que na fase de treinamento utiliza um conjunto de dados que já está classificado. Dessa forma, quando o algoritmo estiver treinado e receber novos dados não classificados, ele consegue predizer qual seria

a classificação desses novos dados. A Figura 16 mostra o fluxo de funcionamento de um aprendizado supervisionado. Um exemplo seria a filtragem de spam, onde o conjunto de dados classificados seriam e-mails rotulados de "spam" ou "não spam" que treinariam um algoritmo para predizer se um novo e-mail é considerado spam ou não.

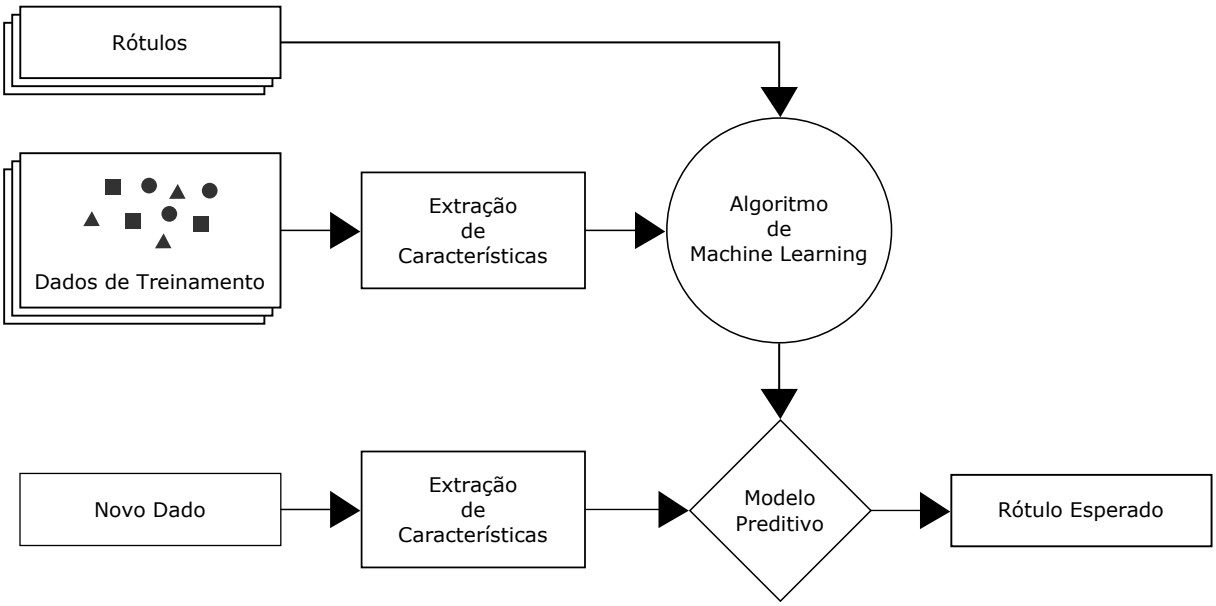


Figura 16: Fluxo do Aprendizado Supervisionado.

O aprendizado não supervisionado recebe de entrada apenas dados não classificados. Logo, o algoritmo precisa identificar similaridades entre os dados de entrada para conseguir classificá-los (MARSLAND, 2014). Um exemplo seria o algoritmo *K-Means*, o qual associa os dados à K centróides, baseado na menor distância euclidiana que cada dado possui em relação a todos os K centróides.

Há também o Aprendizado Profundo - *Deep Learning (DL)* que é chamado dessa maneira devido a sua estrutura que é dividida em diversas camadas. Sendo assim, o DL divide um problema a ser solucionado em diversas partes menores a fim de ir abstraindo e solucionando aos poucos até conseguir solucionar o problema como um todo (BENGIO; COURVILLE, 2016). A próxima subseção irá detalhar melhor algumas das técnicas mais utilizadas em DL.

2.3.1 Rede Neural Artificial

Uma Rede Neural Artificial (RNA) é baseada no sistema nervoso. O sistema nervoso é composto por neurônios biológicos, como ilustra a Figura 17, que se comunicam através de sinapses. Um neurônio biológico recebe impulsos de outros neurônios através de seus dendritos e caso esse impulsos ultrapassem um limiar de ativação, esse neurônio receptor irá gerar outro impulso que será enviado pelo seu axônio (GERSHENSON, 2003).

Dessa maneira, a composição desses neurônios e suas comunicações é o que chamamos de rede neural.

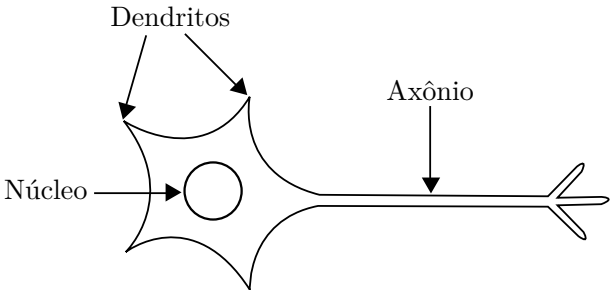


Figura 17: Neurônio biológico.

A RNA também é composta por neurônios e conexões que ligam os mesmos. Entretanto, os neurônios da RNA são neurônios artificiais, como mostrado na Figura 18. Esses neurônios recebem um conjunto de entradas $\vec{x}$ que são multiplicadas respectivamente por um vetor de pesos $\vec{w}$. O resultado do produto escalar entre esses vetores é enviado para uma *função de ativação*. A função de ativação tem o objetivo de definir como a informação desse neurônio irá ser passada adiante, assim uma saída y é gerada.

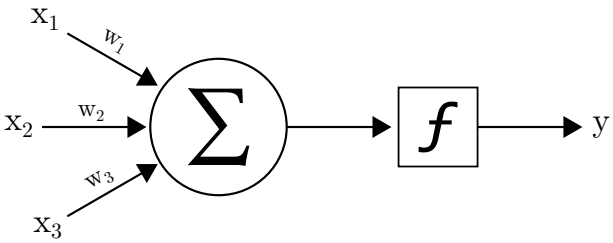


Figura 18: Neurônio artificial.

Os pesos de uma RNA são ajustados para que se obtenha uma saída desejada. Essa etapa de ajuste dos pesos é chamada de *treinamento*. Após treinar uma RNA, ela será capaz de predizer uma saída desejada para um conjunto de entradas ainda não classificadas. As RNA em geral são formadas por diversas camadas de neurônios. A Figura 19 mostra um exemplo de uma RNA com múltiplas camadas. A Perceptron Multicamadas (MLP) é um tipo de RNA que é composta de múltiplas camadas. As camadas podem ser classificadas em camada de entrada, camada oculta ou camada de saída. A camada oculta é chamada dessa maneira devido ao fato de que não é possível acessar os seus neurônios diretamente (DATT, 2012).

2.3.1.1 Rede Neural Convolucional

A Rede Neural Convolucional - *Convolutional Neural Network (CNN)* é um dos tipos de RNA. É chamada dessa maneira porque a rede faz a utilização da função matemática chamada de *convolução* (BENGIO; COURVILLE, 2016). As CNN são bastante

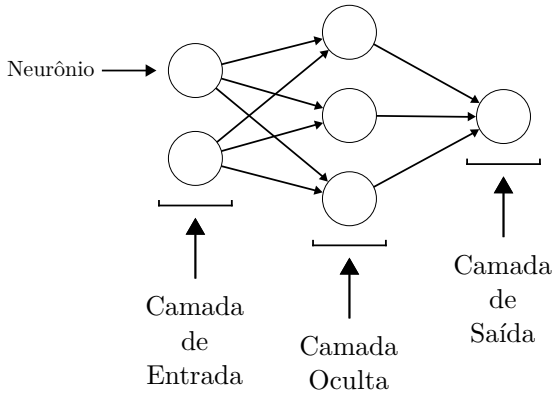


Figura 19: RNA com múltiplas camadas.

utilizadas para reconhecimento de imagens e vídeos. As camadas da CNN são divididas em alguns tipos:

- **Camada convolucional:** essa camada utiliza matrizes de convolução (ou filtros) no qual percorrem toda a imagem. O objetivo desse filtros é extrair algumas características da imagem. As primeiras camadas de convolução abstraem características de baixo nível, como bordas, linhas, cantos da imagem. Conforme a rede tiver mais camadas de convolução, ela consegue abstrair características mais complexas de uma imagem.

- **Camada de Pooling:** essa camada recebe de entrada os mapas de características gerados pela camada convolucional e tem o objetivo de agregar esses valores para que reduza o seu tamanho. Para fazer essa redução, é definida uma máscara que seja menor do que a região total da imagem que irá sendo deslocada por toda a imagem. A cada deslocamento, entre os valores presentes na máscara é considerado apenas o maior valor presente ou a média de todos esses valores. A Figura 20 é um exemplo da camada de pooling que considera o maior valor presente.

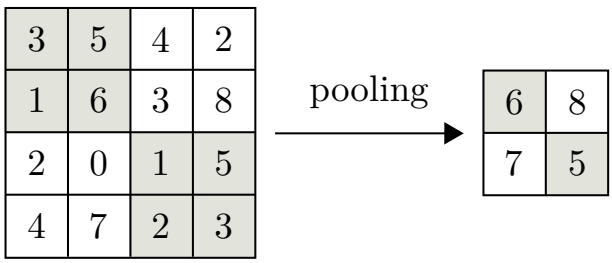


Figura 20: Pooling com uma máscara de tamanho 2×2 que considera o maior valor.

A Figura 21 demonstra a estrutura de uma CNN. Podemos definir a estrutura de uma CNN em duas partes: *extração de características* e *classificação*. Na etapa de extração de características, a cada convolução são gerados novos mapas de características. Com

isso, esses mapas de características passam pela camada de pooling que faz a agregação dos valores, reduzindo o seu tamanho. Uma CNN pode conter uma ou mais camadas de convolução e pooling. Na etapa de classificação, temos a camada densa que possui a característica de seus neurônios serem totalmente interligados. Normalmente, a camada densa é composta de um MLP que por sua vez possui múltiplas camadas.

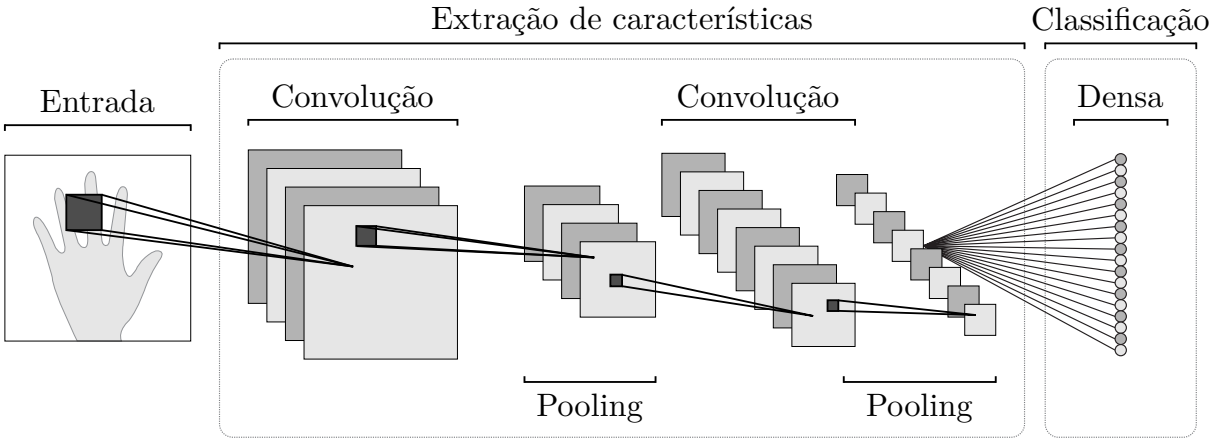


Figura 21: Estrutura de uma CNN.

3 Trabalhos Relacionados

Este Capítulo é destinado a descrever alguns trabalhos que possuem relação com a abordagem selecionada no nosso trabalho.

Li et al. (2011) propuseram o reconhecimento de sinais em American Sign Language (ASL) que é baseado na web e utiliza o Kinect para a captura dos sinais. Os sinais capturados pelo Kinect são pré-processados em um computador pessoal. Os dados pré-processados são enviados para um servidor na web para que os sinais sejam reconhecidos com base em uma biblioteca de sinais predefinidos. O servidor irá comparar o sinal recebido com todos os sinais da biblioteca e nessa comparação o que tiver o menor erro obtido será o sinal candidato. Uma palavra ou frase correspondente ao sinal será a saída caso o sinal seja reconhecido. Os resultados obtidos são descritos como excelentes, porém não é informado detalhes sobre o mesmo.

Zafrulla et al. (2011) utilizam a profundidade da imagem capturada pelo Kinect para fazer reconhecimento da língua de sinais para jogos educacionais desenvolvidos para crianças surdas. Fizeram uma comparação dos resultados obtidos utilizando essa abordagem com o sistema também desenvolvido pelos autores chamado CopyCat, no qual é necessário que o usuário utilize uma luva colorida juntamente com um acelerômetro que rastreia os movimentos das mãos. Fizeram a coleta de 1000 frases em ASL. Seguiram um processo de três etapas, a primeira etapa foi extrair as características das imagens, a segunda etapa foi treinar um *Modelo Oculto de Markov* para cada sinal no vocabulário do jogo, e a última etapa foi testar o modelo com dados independentes. Os resultados obtidos foram uma taxa de 51.5% e 76.12% quando os usuários estavam sentados e em pé, respectivamente. Se comparado com o resultado obtido anteriormente pelo CopyCat de 74.82%, então essa nova abordagem mostra ter um maior potencial.

Anjo, Pizzolato e Feuerstack (2012) utilizaram a informação da profundidade capturada pelo dispositivo Kinect para reconhecer sinais estáticos em Língua Brasileira de Sinais (LIBRAS) em tempo real. Os sinais detectados foram os seguintes grupos de letras do alfabeto: grupo 1 (A, E, I, O, U) e grupo 2 (B, C, F, L, V). O processamento das informações foi dividido nos passos de segmentação e classificação. Para a segmentação foi utilizado um algoritmo desenvolvido, chamado *Virtual Wall*, o qual simula uma parede invisível na frente do rosto e tronco da pessoa para que apenas os braços sejam rastreados. Para a classificação foi utilizado uma Perceptron Multicamadas (MLP) juntamente com outro algoritmo desenvolvido, chamado *ARHCA*, que elimina o braço na imagem recortando apenas a mão. O conjunto de dados utilizado para o treinamento é composto de 250 imagens com recorte de mão e 250 sem o recorte de mão para cada sinal dos grupos

1 e 2, totalizando 5000 amostras. Foram feitos testes utilizando apenas a MLP e outro utilizando a MLP com o algoritmo *ARHCA*. As médias de acertos obtidos no primeiro teste foram de 67.4% para o grupo 1 e 75.4% para o grupo 2. No segundo teste as médias de acertos para os dois grupos foram de 100%.

Agarwal e Thakur (2013) utilizaram o Kinect para fazer reconhecimento de Língua de Sinais (LS). Através de algoritmos de visão computacional, foram detectados os sinais utilizando a profundidade e o movimento das mãos. Foram utilizados no treinamento os sinais dos dígitos de 0 à 9 da LS chinesa Chinese Sign Language (CSL). O repositório de dados utilizado foi o ChaLearn Gesture Dataset (CGD 2011), sendo que dois subconjuntos aleatórios de dados cada um contendo 47 vídeos destinaram-se para o treinamento. Foram gerados arquivos com matrizes das características de cada sinal do conjunto de dados. Esses arquivos foram utilizados em um classificador multiclasse Support Vector Machine (SVM) para gerar modelos de classificação. Os resultados da acurácia obtida para cada subconjunto de dados foram de 92.313% e 90.83%.

Porfirio et al. (2013) fizeram um método de reconhecimento de sinais em LIBRAS que utiliza malhas em 3D e projeções da mão em 2D. As imagens são capturadas pelo dispositivo Kinect. Através da imagem da mão é gerada uma malha em 3D. A vantagem da utilização de malhas em 3D é a habilidade de identificar detalhes nas mãos e nos dedos. Para esse trabalho foi criado um total de 610 vídeos que foram gravados duas vezes por 5 pessoas no qual possui 61 configurações de mão em LIBRAS. Esse conjunto de vídeos foi disponibilizado para uso em outras pesquisas. Os resultados obtidos foram uma taxa de acertos acima de 96%.

Wang et al. (2013) utilizaram um sensor de profundidade para reconhecer gestos feitos em tempo real. Os gestos escolhidos para reconhecimento foram “up”, “down”, “go”, “back” e “click” que são alguns dos comandos utilizados no teclado durante uma apresentação interativa. Para melhorar os resultados obtidos foram utilizados um método para remoção do fundo da imagem e um mapa de distância para detectar as mãos. Para se certificarem de que um gesto está sendo efetuado, foi utilizada uma técnica chamada de *Potential Active Region*, que faz uma seleção dos quadros ativos da imagem, assim enquanto as mãos não forem detectadas não haverá necessidade de processar o trajeto da mão, reduzindo o custo computacional. Também foi proposto um *modelo discriminativo suave* que corrige classificações incorretas dos gestos. Durante a trajetória da mão as imagens vão sendo segmentadas em cada movimento performado. Os movimentos segmentados são classificados através de um SVM. Foram obtidos os resultados de uma taxa de reconhecimento de 90%. Porém, não fizeram o reconhecimento da configuração de mão devido aos gestos utilizados em uma apresentação interativa não dependerem dessa informação.

Almeida, Guimarães e Ramírez (2014) utilizaram da extração de características

em LIBRAS através da estrutura fonológica da língua, como configuração de mão, tipo de movimentação das mãos, pontos de articulação, orientação, entre outros. A captura dos dados foi feita através da ferramenta Kinect. Foram utilizadas técnicas de visão computacional para segmentar e detectar a movimentação das mãos. Para ilustrar a metodologia foi selecionado um conjunto de 34 sinais em LIBRAS. Com isso, foi treinado uma SVM com base nos elementos linguísticos e nas características extraídas. Obtiveram uma acurácia acima de 80%.

Yang (2014) fez o reconhecimento de sinais em tempo real a partir da configuração e movimento de mãos capturados pela ferramenta Kinect. As informações capturadas pelo Kinect foram processadas utilizando o *hierárquico Conditional Random Field (H-CRF)* para reconhecer os sinais a partir da movimentação das mãos. Também foi utilizado *BoostMap* para reconhecer as configurações de mão. Para o treinamento do H-CRF foi utilizado um conjunto de 24 sinais dentre eles 7 sinais de uma mão só e 17 sinais que utilizam as duas mãos. Com isso, obteve-se um resultado de reconhecimento dos sinais de 90.4%.

Bastos, Angelo e Loula (2015) fizeram o reconhecimento de sinais em imagens. As imagens utilizadas foram de um conjunto de 9600 imagens que representam 40 sinais em LIBRAS. Utilizaram dois tipos de descritores de formatos: *Histogram of Oriented Gradients (HOG)* e *Zernike Invariant Moments (ZIM)*. As informações coletadas a partir dos descritores foram utilizadas para treinar uma rede neural MLP em dois estágios. O primeiro estágio foi a extração de informações das imagens utilizando HOG e ZIM e a combinação dessas informações em um vetor de características que foi associado a uma MLP. O segundo estágio foi a utilização de técnicas de processamento de imagem como a de detecção da pele. Para validar os resultados obtidos, foram feitos três testes diferentes: o teste 1 usou um único classificador e em apenas um estágio; o teste 2 usou cada um dos descritores isolados e o teste 3 utilizou um conjunto de 20 imagens de cada sinal que não estavam presentes no conjunto de treinamento. Com isso, os testes mostraram que a maior taxa de reconhecimento das imagens ficou em 96.77% com a utilização dos dois estágios juntos.

Sugianto e Yuwono (2015) desenvolveram uma tecnologia de reconhecimento de sinais para a Indonesian Sign Language (ISL). A captura dos sinais foi feita utilizando o dispositivo Kinect. Para fazer a classificação dos sinais foram utilizadas duas Convolutional Neural Network (CNN) e uma Rede Neural Artificial (RNA). A primeira CNN captura as características da imagem colorida, a segunda CNN captura características da profundidade da imagem e a RNA é usada para classificar as características em classes. São reconhecidos 10 diferentes sinais dinâmicos de um conjunto de 100 sequências de imagens que foram performadas por duas pessoas diferentes. Cada sequência de imagem possui 32 quadros. 80% do conjunto de dados foi destinado para treinamento e 20% foi

destinado para os testes. Para evitar *overfitting* durante o treinamento e generalizar melhor o classificador, foram feitos três casos de treinamento nos quais utilizaram os métodos *data augmentation* e *drop-out*. O primeiro caso foi apenas utilizando o conjunto de treinamento existente. O segundo caso foi utilizado o conjunto de treinamento juntamente com o método drop-out. E o último caso utilizou o conjunto de treinamento e o método *data augmentation*. Os resultados dos testes obtidos de acurácia para os três casos utilizando o próprio conjunto de treinamento foram, respectivamente, de 71.90%, 74.40% e 81.60%. Para o teste com o conjunto de teste a acurácia foi de 73%.

4 Metodologia e Resultados

Neste trabalho temos como objetivo principal fazer o reconhecimento de sinais de uma Língua de Sinais (LS) utilizando imagens de profundidade da mão, e com isso, convertê-las para a escrita de sinais em SignWriting (SW). Como descrito nos Capítulos anteriores, o SW é composto por parâmetros de orientação da palma, configuração de mão, tipos de contatos, movimento, expressão facial e localização corporal. Entretanto, para esse trabalho nos concentramos no parâmetro de configuração de mão. A fim de reconhecer os sinais, treinamos a Convolutional Neural Network (CNN) com um conjunto de imagens de profundidade de 79 configurações de mão e posteriormente com 16 configurações de mão. As imagens de profundidade foram capturadas a partir do dispositivo Kinect.

4.1 Coleta das imagens

Como vimos nos trabalhos relacionados, em sua grande maioria, o sensor de movimentos Kinect foi utilizado como ferramenta de captura dos sinais. Um dos principais motivos da utilização do Kinect é a sua capacidade de capturar não só apenas a imagem em si, mas também a profundidade da imagem, através de um sensor infravermelho. O Kinect também facilita o pré-processamento dos sinais capturados através de um Software Development Kit (SDK) disponível pela empresa desenvolvedora. Devido a esses fatores, fizemos a captura dos sinais utilizando a segunda versão do Kinect em razão de suas especificações técnicas, como a melhoria da qualidade da imagem de profundidade, serem superiores a primeira versão. O grupo de pesquisa Laboratório de Estudos Avançados em Computação (LEA) da UNIPAMPA disponibilizou um dispositivo Kinect para o desenvolvimento do trabalho.

O Kinect fornece a imagem de profundidade em uma resolução de 512×424 pixels e em uma profundidade de até 4,5 metros. Dessa maneira, precisamos extrair apenas a imagem da mão dentro dessa região. Logo, durante a captura das imagens de profundidade, preprocessamos essas imagens eliminando as partes desnecessárias. O SDK do Kinect faz com que as partes do corpo de uma pessoa sejam mapeadas para *articulações de esqueleto*. São no total 25 articulações de esqueleto divididos entre o corpo interno (MICROSOFT, 2015). Utilizamos essa informação para rastrear as articulações do esqueleto representantes das mãos de uma pessoa fazendo os sinais. Com isso, localizamos a posição das mãos e removemos o fundo e qualquer outra informação desnecessária na imagem. Utilizamos o sensor de profundidade do Kinect para conseguir obter apenas as mãos na imagem. Optamos por capturar a imagem na resolução de 54×54 pixels. Um dos problemas enfrentados foi conseguir manter essa resolução para diferentes tamanhos

de mãos. Dessa maneira, fizemos algumas tentativas de calcular o tamanho da mão e fazer o recorte com base nessa medida. Porém, o Kinect não tinha muita estabilidade para calcular o tamanho da mão em diferentes configurações de mão. Com isso, definimos previamente um tamanho fixo de 54×54 pixels. Além disso, determinamos que a cor mais clara na imagem simboliza uma profundidade maior, e a cor mais escura uma profundidade menor. A Figura 22 mostra o resultado da imagem preprocessada.

Figura 22: Imagem capturada pelo Kinect e preprocessada. Os tons de cinza mais claros correspondem à pixels mais próximos do observador.

Foi desenvolvida uma aplicação na linguagem C# para fazer o preprocessamento descrito no parágrafo anterior. Essa aplicação também foi construída para auxiliar na coleta de imagens. Na Figura 23 podemos observar a tela da aplicação em funcionamento.

Figura 23: Tela da aplicação desenvolvida para a captura e preprocessamento das imagens.

4.1.1 Imagens capturadas

Utilizando a aplicação desenvolvida, conseguimos 5 pessoas voluntárias para capturar as imagens necessárias. A cada uma dessas pessoas foi solicitado executar 79 configurações de mão diferentes. As configurações de mão eram mostradas na aplicação e os voluntários iam repetindo uma por vez. Desse modo, coletamos um total de 395 imagens de profundidade da mão. Cada imagem possui a resolução de 54×54 pixels em escala de cinza.

4.2 Treinamento e validação

Com as imagens capturadas, iniciamos o processo de treinamento da CNN. Por questões de robustez, foi utilizada a biblioteca *Keras*[*] de Redes Neurais Artificiais (RNAs) desenvolvida em Python.

A RNA como foi descrita no Capítulo 2, necessita de dados para que possa aprender a reconhecer padrões. Logo, as imagens com os sinais são as entradas que a RNA precisa para poder fazer o reconhecimento do mesmo. A razão pela qual optamos por utilizar a CNN é porque ela já foi utilizada para o reconhecimento de objetos em imagens e apresentou resultados bastante promissores como no trabalho de Sugianto e Yuwono (2015).

4.2.1 Parâmetros

A CNN possui alguns parâmetros de treinamento que podem ser modificados a fim de melhorar o seu aprendizado. Dessa forma, em nosso trabalho testamos diferentes configurações para conseguir aprimorar o resultado final. Os parâmetros modificados foram: o número de camadas de convolução, número de camadas de pooling, número de camadas densas, número de neurônios nas camadas densas, o tamanho da matriz de convolução. Para os treinamentos foram definidos 32 filtros para todas as camadas de convolução. O parâmetro de número de épocas foi definido como 50 para todos os treinamentos, visto que a partir da época 50 não ocorreram grandes mudanças nos resultados.

A função de ativação utilizada no treinamento foi a Retified Linear Unit (ReLU) que é uma função não saturante $f(x) = max(0, x)$. De acordo com Krizhevsky, Sutskever e Hinton (2012), a utilização da função ReLU torna o treinamento da CNN mais rápido em relação a utilização de funções saturantes como a função sigmoid $f(x) = (1 + e^{-x})^{-1}$. Contudo, na última camada da RNA foi utilizada a função de ativação *softmax* que tem o objetivo de atribuir probabilidades de que uma dada entrada da RNA seja pertencente a

[*]Disponível em: http://keras.io/

cada uma das classes de saída. Sendo assim, a soma das probabilidades de que a imagem pertença a cada classe deve ser 1. A função *softmax* é dada por:

$$softmax(x)_i = \frac{\exp(x_i)}{\sum_j \exp(x_j)} \tag{4.1}$$

Com relação a Equação 4.1, o i representa a classe que está sendo calculada e j representa cada uma das classes.

Para o medir os erros durante o treinamento, foi utilizada a função de custo *cross-entropy (CE)* que é dada por:

$$CE = -\sum_{i=1}^{n}[(1-d_i)\log_2(1-a_i) + d_i \log_2(a_i)] \tag{4.2}$$

Considerando a Equação 4.2, a_i representa o valor de saída no neurônio i, d_i representa o valor desejado de saída no neurônio i, e n representa o número de saídas. Para minimizar os erros, foi utilizado o otimizador *ADADELTA* que não necessita de ajuste manual da taxa de aprendizagem (ZEILER, 2012).

4.2.2 Treinamentos executados e Resultados obtidos

Para iniciar o treinamento, definimos que do conjunto total de 395 imagens, utilizaremos 316 imagens para o treinamento da CNN e 79 imagens para o teste de validação da CNN.

Desse modo, treinamos a CNN para tentar reconhecer 79 configurações de mão diferentes. Utilizamos diferentes combinações de parâmetros no intuito de obter os melhores resultados. A Figura 24 apresenta um gráfico com os resultados do treinamento.

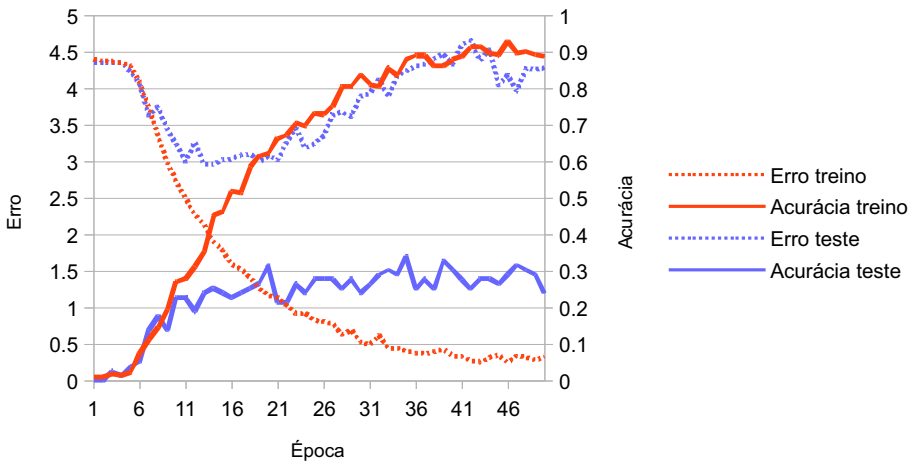


Figura 24: Resultados do treinamento com 79 configurações de mão.

Observando o gráfico, podemos perceber que a CNN atingiu uma acurácia de 90% no conjunto de treino. Entretanto, para o conjunto de testes, a taxa de acerto em 24%. Isso representa um cenário de *overfitting*, onde a CNN memorizou o conjunto de treinamento e com isso, não conseguiu generalizar o suficiente para acertar novas entradas não pertencentes ao conjunto de treinamento. O principal motivo desse acontecimento se deve ao fato de que não conseguimos obter amostras de imagens suficientes para generalizar as 79 configurações de mão diferentes de maneira eficiente.

Dessa maneira, decidimos então focar o trabalho no reconhecimento de um conjunto menor de 16 configurações de mão diferentes. A Figura 25 mostra as 16 configurações previamente selecionadas.

Figura 25: 16 configurações de mão definidas.

Com a redução do conjunto de configurações de mão, tivemos que eliminar as amostras de imagens que não pertenciam a essas 16 configurações de mão. Nesse caso, ficamos com um total de 80 imagens. Novamente, partimos as nossas imagens em dois conjuntos. Para o conjunto de treinamento ficamos com 64 imagens e para o conjunto de testes ficamos com 16 imagens.

Assim, treinamos novamente a CNN com esse outro conjunto de amostras. Utilizando três camadas de convolução, três camadas de pooling e uma camada densa com 128 neurônios e obtivemos uma acurácia de 37.5% do conjunto de testes. O problema do *overfitting* continuou ocorrendo. Com isso, para tentar reduzir o *overfitting* modificamos a CNN para utilizar a técnica do *dropout*. Essa técnica consiste basicamente em escolher aleatoriamente com uma probabilidade p de que cada neurônio da RNA seja removido temporariamente, prevenindo que a RNA se adapte ao conjunto de treinamento (SRIVASTAVA et al., 2014). Dessa maneira, a CNN conseguiu atingir a acurácia de 62% do conjunto de testes.

Para aprimorar os resultados, decidimos coletar mais imagens das 16 configurações de mão definidas. Entretanto, para cada uma dessas configurações de mão, foi feita a coleta da configuração em 5 ângulos diferentes. A coleta em diferentes ângulos faz com que a CNN consiga abstrair melhor variações nas configurações de mão. A Figura 26

ilustra uma das amostras dessa coleta.

Figura 26: Configuração de mão em 5 ângulos diferentes.

Com a nova coleta de 5 novas imagens para cada configuração de mão, acrescentamos 80 imagens ao nosso conjunto de treinamento original, totalizando 144 imagens para o treinamento. Refizemos o treinamento com os mesmos parâmetros definidos anteriormente e a CNN conseguiu atingir a acurácia de 81% do conjunto de testes. A Figura 27 apresenta o gráfico do treinamento.

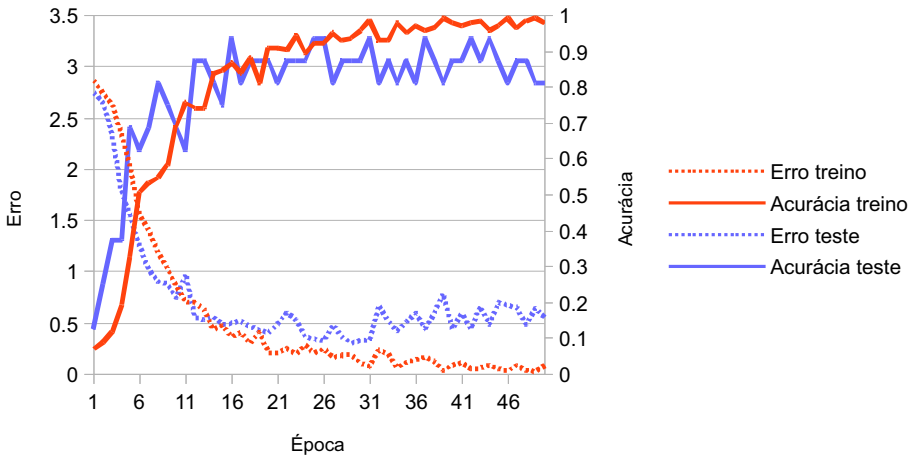


Figura 27: Resultados do treinamento com 144 configurações de mão.

Na Figura 27 é possível observar que o erro do teste tem uma convergência maior, chegando ao valor de 0.5582 na última época. Além disso, até a época 20, houve uma redução drástica dos erros. Após a época 20, o erro do teste começou a oscilar e aumentar gradualmente.

Conseguimos alcançar uma acurácia relativamente alta, porém, insistimos em novos treinamentos para tentar buscar resultados ainda superiores. Desse modo, aumentamos a quantidade de neurônios na camada densa da CNN para 256 neurônios e obtivemos resultados de 87.5% de acurácia. O gráfico do último treinamento é apresentado na Figura 28.

A arquitetura da CNN para o último treinamento pode ser visualizada na Figura 29. A entrada é composta de imagens de dimensão 54×54 pixels. As imagens passam por uma camada de convolução com 32 filtros com o tamanho de 3×3, logo após

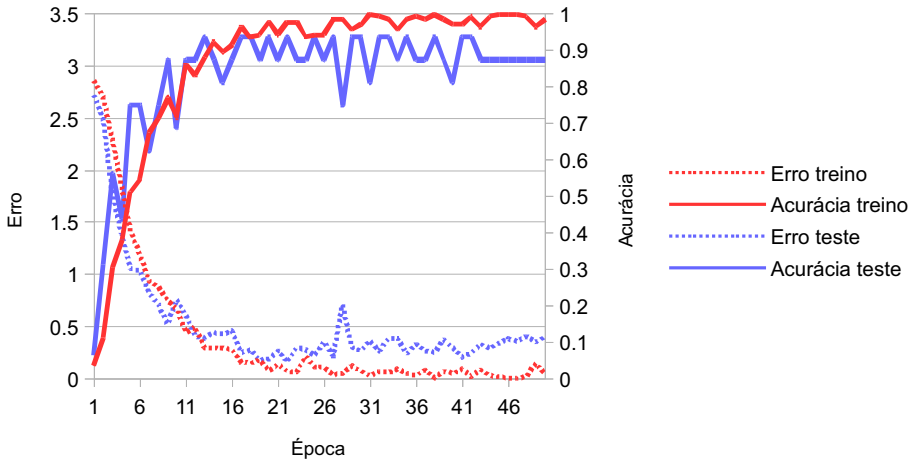


Figura 28: Resultados do treinamento com 256 neurônios.

a convolução são gerados 32 mapas de características com o tamanho de 52×52. Os 32 mapas de características passam por uma camada de max-pooling com filtros de tamanho 2×2, reduzindo os mapas para 26×26. Ocorre uma segunda convolução com 32 filtros de tamanho 3×3, gerando 32 novos mapas de características do tamanho 24×24. Os mapas de características passam novamente por uma camada de max-pooling com filtros de tamanho 2×2, reduzindo os mapas para 12×12. Os 32 mapas de características são passados para uma terceira camada de convolução com as mesmos parâmetros anteriores, gerando 32 novos mapas de características com o tamanho de 10×10. Os mapas são reduzidos pela terceira camada de max-pooling, ficando com a resolução de 5×5. O *dropout* é aplicado com uma probabilidade $p = 0.25$. Os valores são colocados como entrada na camada densa com 256 neurônios. Outro *dropout* é aplicado com probabilidade $p = 0.5$. Por fim, os valores são passados para a camada de saída com 16 neurônios que distribuem a probabilidade de que o valor de entrada da CNN pertença a uma das 16 classes de configurações de mão.

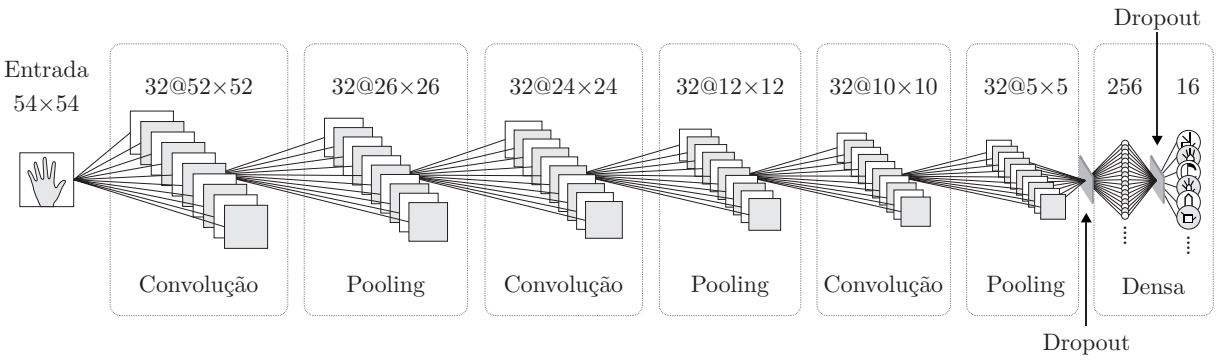


Figura 29: Arquitetura final da CNN.

4.3 Conversão para SignWriting

Após o reconhecimento da configuração de mão pela CNN, precisamos converter essa configuração para a escrita em SW. Para isso, utilizamos do Formal SignWriting (FSW) para representar o símbolo em SW. A Tabela 2 apresenta as configurações de mão em paralelo com o símbolo em SW e o parâmetro *symbol key* do FSW.

LIBRAS	SW	Symbol Key
		S15a00
		S15d00
		S14700
		S14400
		S14c00
		S16c00
		S16d00
		S11500

LIBRAS	SW	Symbol Key
		S1dc00
		S12d00
		S11e00
		S10000
		S10e00
		S19c00
		S19a00
		S1f500

Tabela 2: Configurações de mão em SW e FSW

Com a CNN treinada, ou seja, com os pesos sinápticos ajustados, a conversão da configuração de mão para a escrita em SW é feita com a construção de uma expressão regular no formato do FSW. Como visto no Capítulo 2, o FSW é um conjunto de caracteres ASCII que representam sinais logográficos. Além disso, o FSW é dividido em alguns conjuntos discutidos anteriormente. A área definida para a escrita do sinal (*SignBox*) foi definida com o valor fixo de "M512x424" que indica uma área com a resolução de 512×424, a mesma resolução de imagem de profundidade fornecida pelo Kinect. Para o conjunto *Symbol Key* o valor é definido conforme os valores da Tabela 2. Para o conjunto *Coordinates* foi definido o valor fixo de "000x000" que indica que o símbolo irá aparecer na posição do topo mais a esquerda da *SignBox*. Com os valores dos conjuntos definidos, resta apenas concatená-los formando uma expressão regular no formato do FSW.

5 Conclusão e Trabalhos Futuros

Nesse trabalho tivemos como objetivo principal fazer o reconhecimento automático de sinais de uma Língua de Sinais (LS) para fazer a escrita de sinais na notação de escrita em SignWriting (SW). Fomos motivados pela necessidade de desenvolver uma maneira mais fácil de converter os sinais para a escrita de sinais. Mas também percebemos a ausência de trabalhos que abordassem esse tipo de conversão na literatura. Para alcançar esse objetivo, propomos a utilização de Rede Neural Artificial (RNA) que é um dos modelos de aprendizado de máquina supervisionado. Além disso, decidimos utilizar a Convolutional Neural Network (CNN) que é um dos tipos de RNA e que se mostrou viável para o reconhecimento de imagens. Um dos principais motivos da viabilidade no uso de CNN em reconhecimento de imagens se deve ao fato de que as imagens possuem uma estrutura espacial que a CNN consegue explorar através da utilização de matrizes de convoluções que percorrem toda a imagem e conseguem fazer a extração de características.

Foi feita a captura de imagens de profundidade de 79 configurações de mão por meio do dispositivo Kinect. Para auxiliar na captura das imagens foi desenvolvida uma aplicação que captura e preprocessa as imagens obtidas pelo Kinect, fazendo o recorte da mão e eliminando o fundo e qualquer outra informação desnecessária na imagem.

Durante a fase de treinamento, começamos treinando a CNN para reconhecer 79 configurações e mão. Entretanto, os resultados não foram como o esperado, obtendo apenas uma acurácia de 24% do conjunto de testes. Percebemos que não seria possível reconhecer um conjunto muito grande de configurações de mão utilizando poucas amostras que conseguimos coletar. Dessa maneira, decidimos considerar apenas um subconjunto de 16 configurações de mão. No primeiro treinamento com o subconjunto obtivemos uma acurácia de 37.5%, que já ultrapassa o nosso primeiro treinamento. Porém, observamos a ocorrência de *overfitting* que estava limitando o desempenho da CNN. Com a introdução do método *dropout* no treinamento, conseguimos atingir uma acurácia de 62% do conjunto de testes. Para elevar ainda mais os nosso resultados resolvemos fazer novas capturas de imagens com ângulos diferentes, para que a CNN consiga abstrair melhor as configurações de mão. Com isso, obtivemos uma acurácia de 81% do conjunto de testes. Por fim, modificamos a camada densa para 256 neurônios e alcançamos uma acurácia de 87.5%. Com a CNN treinada, fizemos a conversão da configuração de mão para a escrita em SW, através da utilização da expressão regular Formal SignWriting (FSW).

Concluímos que através do treinamento da CNN com imagens de profundidade de configurações de mão é possível obter resultados satisfatórios. Contudo, é necessário fazer uma coleta de um volume grande de imagens, dessa maneira, é possível evitar o

overfitting. Além disso, encontrar os parâmetros mais adequados da CNN ainda é um desafio. Conseguimos atingir uma acurácia superior ao trabalho de Sugianto e Yuwono (2015) que foi de 73%.

Como trabalhos futuros, sugerimos uma coleta maior de imagens, principalmente com um número maior de pessoas. Dessa maneira, seria possível treinar um CNN para identificar uma quantidade maior de configurações de mão. Além disso, também sugerimos explorar o reconhecimento dos outros parâmetros da LS, como a orientação da palma e o movimento.

Referências

AGARWAL, A.; THAKUR, M. Sign language recognition using microsoft kinect. In: *Contemporary Computing (IC3), 2013 Sixth International Conference on*. [S.l.: s.n.], 2013. p. 181–185. Citado na página 36.

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. Citado na página 36.

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*. [S.l.], 2012. p. 259–268. Citado na página 35.

BARRETO, M.; BARRETO, R. Escrita de sinais sem mistérios. *Ed. do Autor, BH*, 2012. Citado 2 vezes nas páginas 21 e 24.

BARROS, M. E. et al. Elis-escrita das línguas de sinais: proposta teórica e verificação prática. Florianópolis, SC, 2008. Citado 3 vezes nas páginas 9, 23 e 24.

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*. [S.l.], 2015. p. 305–312. Citado na página 37.

BENGIO, I. G. Y.; COURVILLE, A. Deep learning. Book in preparation for MIT Press. 2016. Disponível em: <http://www.deeplearningbook.org>. Citado 3 vezes nas páginas 29, 30 e 31.

CIVIL, C. *LEI Nº 10.436, DE 24 DE ABRIL DE 2002*. 2002. Disponível em: <http://www.planalto.gov.br/ccivil_03/leis/2002/L10436.htm>. Citado 2 vezes nas páginas 17 e 20.

COSSIO, M. L. T. et al. LIBRAS - Língua Brasileira de Sinais. *Uma ética para quantos?*, XXXIII, n. 2, p. 81–87, 2012. ISSN 0717-6163. Citado 2 vezes nas páginas 9 e 21.

DATT, G. An evolutionary approach : Analysis of artificial neural networks. 2012. Disponível em: <http://www.ijetae.com/files/Volume2Issue1/IJETAE_0112_30.pdf>. Citado na página 31.

DIZEU, L. C. T. D. B.; CAPORALI, S. A. A língua de sinais constituindo o surdo como sujeito. *Educação & Sociedade*, v. 26, n. 91, p. 583–597, 2005. ISSN 0101-7330. Citado na página 19.

GERSHENSON, C. Artificial neural networks for beginners. *arXiv preprint cs/0308031*, 2003. Citado na página 30.

GUIMARÃES, C.; GUARDEZI, J. F.; FERNANDES, S. Sign language writing acquisition–technology for a writing system. In: IEEE. *System Sciences (HICSS), 2014 47th Hawaii International Conference on*. [S.l.], 2014. p. 120–129. Citado na página 24.

HANKE, T. Hamnosys-representing sign language data in language resources and language processing contexts. In: *LREC*. [S.l.: s.n.], 2004. v. 4. Citado 3 vezes nas páginas 9, 22 e 23.

HONORA, M.; FRIZANCO, M. L. E. Livro ilustrado de língua brasileira de sinais: desvendando a comunicação usada pelas pessoas com surdez. *São Paulo: Ciranda Cultural*, 2009. Citado na página 20.

INES, G. do. *Grupo de Pesquisa do curso de LIBRAS do Instituto Nacional de Educação*. 2011. Citado 2 vezes nas páginas 9 e 20.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2012. p. 1097–1105. Citado na página 41.

LI, K. F. et al. A web-based sign language translator using 3d video processing. In: IEEE. *Network-Based Information Systems (NBiS), 2011 14th International Conference on*. [S.l.], 2011. p. 356–361. Citado na página 35.

MARSLAND, S. *Machine learning: an algorithmic perspective*. [S.l.]: CRC press, 2014. Citado na página 30.

MARTIN, J. A linguistic comparison: Two notation systems for signed languages: Stokoe notation and sutton signwriting. *Unpublished manuscript, Western Washington University*, 2000. Citado na página 22.

MICROSOFT. *Developing with Kinect for Windows*. 2015. Disponível em: <https://dev.windows.com/en-us/kinect/develop>. Citado na página 39.

MITCHELL, T. M. *Machine learning. WCB*. [S.l.]: McGraw-Hill Boston, MA:, 1997. Citado na página 29.

ORGANIZATION, W. H. Deafness and hearing loss. 2015. Disponível em: <http://www.who.int/mediacentre/factsheets/fs300/en/>. Citado na página 17.

OVIEDO, A. Roch Ambroise Auguste Bebian. p. 1–6, 2007. Disponível em: <http://www.cultura-sorda.eu/resources/Roch_Ambroise_Auguste_Bebian.pdf>. Citado 2 vezes nas páginas 11 e 22.

PANSE, S.; GROMISCH, E. S. *Learn American Sign Language: Stokoe Notation*. 2012. Disponível em: <www.brighthubeducation.com/special-ed-hearing-impairments/50514-stokoe-notation-and-american-sign-language/>. Citado na página 22.

PEREIRA, G. K. LIBRAS: Língua Brasileira de Sinais. 2010. Citado 2 vezes nas páginas 19 e 20.

PERLMUTTER, D. M. What is Sign Language ? *Linguistic Society of America*, v. 6501, n. 202, 2013. Citado na página 19.

PORFIRIO, A. J. et al. Libras sign language hand configuration recognition based on 3d meshes. In: IEEE. *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*. [S.l.], 2013. p. 1588–1593. Citado na página 36.

QUADROS, R. M. *Um capítulo da história do SignWriting*. 2005. Citado na página 25.

SANTOS, M. d. F.; GODOI, P.; SILVA, V. F. da. Língua Brasileira de Sinais no Contexto Bilingue. 2013. Disponível em: <http://monografias.brasilescola.com/educacao/lingua-brasileira-sinais-no-contexto-escola-bilingue.htm>. Citado na página 20.

SLEVINSKI, S. The SignPuddle Standard for SignWriting Text. 2015. ISSN 1098-6596. Citado 2 vezes nas páginas 28 e 29.

SRIVASTAVA, N. et al. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, JMLR. org, v. 15, n. 1, p. 1929–1958, 2014. Citado na página 43.

STUMPF, M. R. Língua de sinais: escrita dos surdos na internet. In: *V Congresso Ibero-Americano de Informática na Educação–RIBIE–Chile*. [S.l.: s.n.], 2000. Citado na página 24.

STUMPF, M. R. Aprendizagem de escrita de língua de sinais pelo sistema signwriting: língua de sinais no papel e no computador. 2005. Citado 3 vezes nas páginas 9, 23 e 24.

SUGIANTO, N.; YUWONO, E. I. Indonesian dynamic sign language recognition at complex background with 2d convolutional neural networks. In: . [S.l.: s.n.], 2015. Citado 3 vezes nas páginas 37, 41 e 48.

SUTTON, V. The Importance of WritingSign Language Down In Columns. 1998. Citado 2 vezes nas páginas 26 e 27.

SUTTON, V. Signwriting. In: . [S.l.: s.n.], 2009. Citado 4 vezes nas páginas 9, 25, 26 e 27.

WANG, H. et al. Depth sensor assisted real-time gesture recognition for interactive presentation. *Journal of Visual Communication and Image Representation*, Elsevier, v. 24, n. 8, p. 1458–1468, 2013. Citado na página 36.

YANG, H.-D. Sign language recognition with the kinect sensor based on conditional random fields. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 15, n. 1, p. 135–147, 2014. Citado na página 37.

ZAFRULLA, Z. et al. American sign language recognition with the kinect. In: ACM. *Proceedings of the 13th international conference on multimodal interfaces*. [S.l.], 2011. p. 279–286. Citado na página 35.

ZEILER, M. D. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012. Citado na página 42.