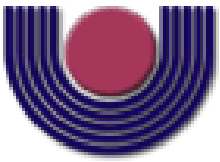

UNIOESTE – Universidade Estadual do Oeste do Paraná

CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
Colegiado de Ciência da Computação

Curso de Bacharelado em Ciência da Computação

Uma arquitetura baseada em redes neurais convolucionais para reconhecimento de sinais da Libras

Renan Tashiro

CASCAVEL
2018

RENAN TASHIRO

UMA ARQUITETURA BASEADA EM REDES NEURAIS CONVOLUCIONAIS PARA RECONHECIMENTO DE SINAIS DA LIBRAS

Monografia apresentada como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação, do Centro de Ciências Exatas e Tecnológicas da Universidade Estadual do Oeste do Paraná - Campus de Cascavel

Orientador: Prof. Dr. Adair Santa Catarina

CASCAVEL
2018

RENAN TASHIRO

UMA ARQUITETURA BASEADA EM REDES NEURAIS CONVOLUCIONAIS PARA RECONHECIMENTO DE SINAIS DA LIBRAS

Monografia apresentada como requisito parcial para obtenção do Título de *Bacharel em Ciência da Computação*, pela Universidade Estadual do Oeste do Paraná, Campus de Cascavel, aprovada pela Comissão formada pelos professores:

_____
Prof. Dr. Adair Santa Catarina (Orientador)
Colegiado de Ciência da Computação, UNIOESTE

_____
Me. Vanderlize Dalgalo (Coorientadora)
Programa de Educação Especial - PEE, UNIOESTE

_____
Prof. Dr. André Luiz Brun
Colegiado de Ciência da Computação, UNIOESTE

_____
Prof. Dr. Carlos José Maria Olguín
Colegiado de Ciência da Computação, UNIOESTE

Cascavel, 31 de outubro de 2018

Dedico este trabalho aos futuros leitores.

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

A.C.T.	Acurácia do Conjunto de Treinamento
API	*Application Programming Interface*
Libras	Língua Brasileira de Sinais
LRCN	*Long-short Recurrent Convolutional Network*
ReLU	*Rectified Linear Unit*
RNA	Redes Neurais Artificiais
RNC	Redes Neurais Convolucionais
S3D	*Spatiotemporal-separable 3D Convolutions*
TCLE	Termo de Consentimento Livre e Esclarecido
TILS	Tradutor e Intérprete da Língua de Sinais
T.T.	Tempo de treinamento
Unioeste	Universidade Estadual do Oeste do Paraná

Sumário

Resumo

No Brasil, de acordo com o último Censo Demográfico (2012) cerca de 9,7 milhões de pessoas têm algum tipo de deficiência auditiva, sendo que 2,1 milhões possuem deficiência auditiva severa. A comunicação entre surdos e os ouvintes que não são usuários da Libras dá-se, muitas vezes, através de um intermediário que seja Tradutor e Intérprete da Língua de Sinais. Porém, quando não há esse intermediário a comunicação pode ser comprometida, afetando a qualidade de vida de pessoas que dependem da Libras para se comunicar. Neste trabalho se construiu uma base de dados contendo vídeos de voluntários sinalizando sinais da Libras, para então treinar uma rede neural convolucional com objetivo de classificar esses sinais, visando servir como parte de um sistema tradutor de Libras-Português. Como resultado, a base de dados possui 48 sinais e cerca de 34.560 vídeos. Uma acurácia de aproximadamente 50% foi alcançada ao treinar a rede neural para classificar todos os sinais, considerando a métrica *top-5*. Melhorias na acurácia devem ser procuradas, mas os resultados mostraram serem promissores visando à construção futura de um sistema tradutor Libras-Português.

Palavras-chave: tradução, base de dados, redes neurais, Libras.

Capítulo 1
Introdução

A Língua de Sinais é o principal meio de comunicação entre pessoas com deficiência auditiva. Segundo a Lei nº 10.436, de 24 de abril de 2002, a Língua Brasileira de Sinais (Libras) é reconhecida como a língua oficial para a comunidade surda, constituída por um sistema linguístico de natureza visual-motora com estrutura gramatical própria [BRASIL, 2002]. De acordo com o último Censo Demográfico cerca de 9,7 milhões de pessoas têm algum tipo de deficiência auditiva, sendo que 2,1 milhões possuem deficiência auditiva severa [IBGE, 2012].

A comunicação entre surdos e os ouvintes que não são usuários da Libras dá-se, muitas vezes, através de um intermediário o Tradutor e Intérprete da Língua de Sinais (TILS). Porém, em muitos casos, não se dispõe desse profissional; é o caso da Universidade do Estadual Oeste do Paraná (Unioeste) que, no início deste ano, apresentou escassez desses profissionais para atender à comunidade acadêmica dos *campi* dessa instituição. A carência de profissionais TILS e o desconhecimento da Libras criam dificuldades para a comunicação entre surdos e ouvintes, motivando a busca por soluções automatizadas que, em última instância, façam a tradução Libras-Português-Libras.

Em 2012, uma arquitetura de redes neurais convolucionais (RNC), chamada *AlexNet*, ganhou uma competição de classificação de imagens em alta definição: a *ImageNet* LVSRC-2012 [KRIZHEVSKY, SUTSKEVER e HINTON, 2012]. Nesta competição as imagens estão divididas em 1000 classes diferentes de imagens e a *AlexNet* conseguiu um desempenho significativamente superior a outros métodos não baseados em redes neurais artificiais (RNA), com um *top-5 error* de 15.3%. O segundo melhor resultado foi 26.2%. Isso gerou uma grande atenção para as RNC e desde então elas vêm sendo exploradas em diversas outras aplicações, como a classificação de vídeos como é mostrado por Donahue *et al*. (2017) que, em seu trabalho, apresentam uma classe de arquiteturas de RNC denominada *Long-short Recurrent Convolutional Networks* (LRCN) aplicada em três problemas diferentes: detecção

de gestos, descrição do conteúdo de imagens e de vídeos.

Os bons resultados obtidos com diferentes arquiteturas de RNC, em tarefas envolvendo a classificação de imagens e vídeos, motivaram a realização deste trabalho, que tem como foco explorar o uso de RNC no reconhecimento dos sinais da Libras, uma das etapas para a construção de um sistema de tradução de Libras para a Língua Portuguesa.

1.1 Objetivos

Este trabalho teve dois objetivos principais; o primeiro deles foi criar uma base de dados que contivesse vídeos com pessoas sinalizando sinais da Libras; o segundo foi avaliar uma arquitetura de RNC na tarefa de reconhecimento dos sinais da Libras, usando como entrada os vídeos desta mesma base de dados.

Para atender ao primeiro objetivo foi criada uma base de dados ampla para auxiliar na pesquisa e avaliação de métodos para tradução dos sinais da Libras para Língua Portuguesa. Essa base tem um conjunto significativo de sinais permitindo a criação de diálogos completos; ao mesmo tempo esses sinais apresentam similaridades entre si, porém com significados distintos, no intuito de serem úteis para avaliação da sensibilidade dos métodos utilizados no reconhecimento dos sinais.

Visando contemplar o segundo objetivo foi realizada uma análise para identificar quais fatores afetam o desempenho da RNC na tarefa de reconhecer os sinais da Libras. Especificamente, o processo de reconhecimento utiliza como entrada um segmento de vídeo e retorna seu significado ou uma palavra equivalente na Língua Portuguesa.

1.2 Organização do Documento

O restante deste documento está organizado da seguinte forma. O Capítulo 2 apresenta a revisão de literatura sobre os elementos essenciais para a realização deste trabalho: a Libras e algumas considerações sobre a implementação de um tradutor de Libras para Língua Portuguesa, os fundamentos de RNA, as características das RNC usadas no processo de classificação de imagens e vídeos e os trabalhos correlatos, que visam à aplicação de RNA e outros métodos no reconhecimento de sinais. O Capítulo 3 descreve a metodologia adotada na realização deste trabalho, incluindo como foi coletada e construída a base de dados utilizada no treinamento da rede neural desenvolvida para reconhecer sinais da Libras, além de quais ações e ferramentas foram adotadas na implementação, treinamento e teste da rede neural.

O Capítulo 4 apresenta e discute os resultados obtidos. Por fim, o Capítulo 5 traz as considerações finais sobre o trabalho realizado.

Capítulo 2
Fundamentação Teórica

2.1 Libras

A Língua Brasileira de Sinais (Libras), de acordo com a Lei nº 10.436 de 24 de abril de 2002, é a língua de sinais oficial no Brasil [BRASIL, 2002]. Ela é caracterizada pelo uso de gestos corporais e faciais, que denominam um sinal, para expressar significados e sentimentos. Libras é uma das várias línguas de sinais existentes no mundo e possui uma estrutura linguística própria e independente da Língua Portuguesa. Ela é utilizada primariamente pela comunidade surda brasileira como principal meio de comunicação.

2.1.1 Características da Libras

Vale ressaltar que, assim como a Língua Portuguesa, a Libras possui variações dependendo da região ou a condição social de quem a utiliza; diferentes sinais podem ser utilizados para representar um mesmo significado. A Libras também é uma língua dinâmica, que muda com o tempo [STROBEL e FERNANDES, 1998]. Os sinais da Libras possuem um conjunto de cinco parâmetros que os caracterizam e os principais são [BRITO, QUADROS e FELIPE, 2017]:

- A mão assume uma **configuração** ou uma forma, quando um sinal é realizado. De acordo com o sinal a ser realizado, uma das mãos ou ambas as mãos devem ser configuradas adequadamente. A Figura 2.1 mostra algumas configurações de mãos presentes na Libras;
- Os sinais são executados em uma região do corpo ou no espaço neutro, denominado **ponto de articulação**. Por exemplo, o sinal correspondente a ESQUECER é realizado tocando uma parte do corpo, precisamente a testa; já o sinal que corresponde a QUENTE é realizado à frente ao queixo;

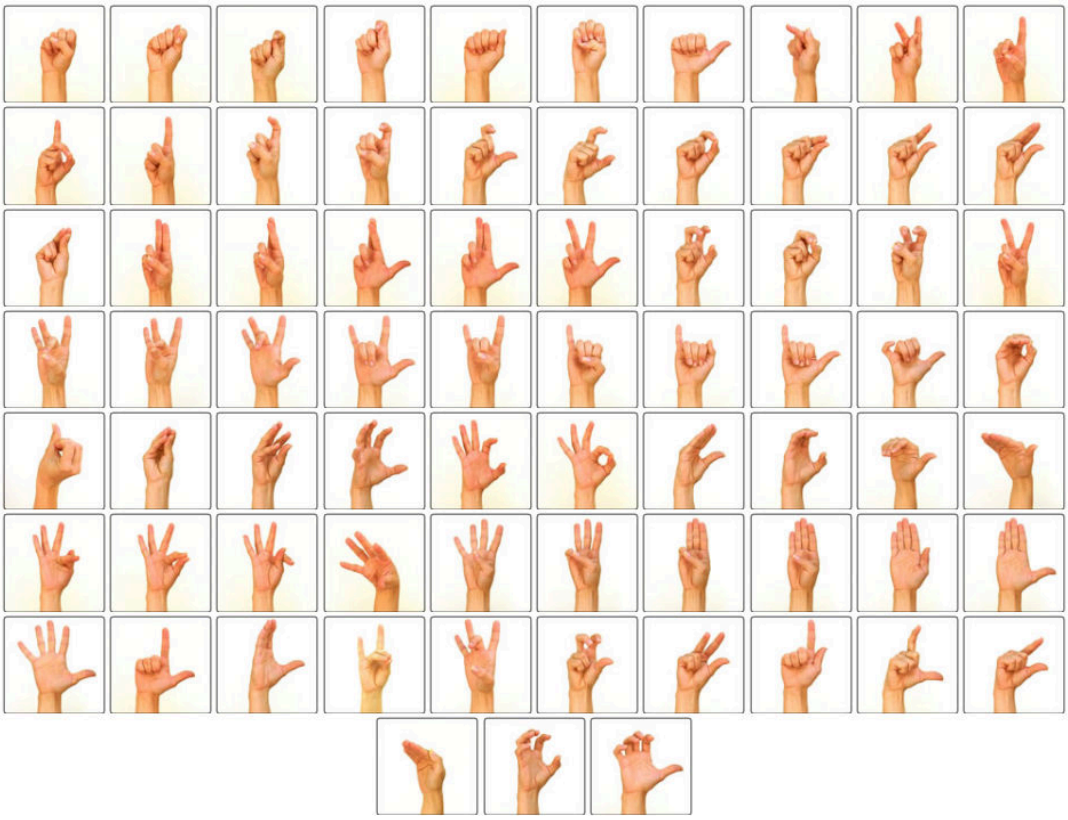

Figura 2.1: Configurações de mão na Libras
Fonte: ACESSIBILIDADE BRASIL (2018)

- Os sinais podem ter ou não um **movimento**. Os sinais correspondentes às letras do alfabeto são em sua maioria estáticos, enquanto outros sinais geralmente possuem algum tipo de movimento associado;

- As palmas devem apontar em uma direção, definindo a **orientação das mãos**. Um sinal pode ter a palma da mão apontada para o emissor, como no caso do sinal MEU-NOME, ou apontada para o receptor, como no caso do sinal SEU-NOME. No caso desses sinais, apenas o parâmetro orientação das mãos é diferente. Vale ressaltar que a palma da mão pode estar apontada para qualquer direção e não necessariamente para o emissor ou para o receptor do sinal;

- Uma **expressão facial** é um componente não manual que pode ser utilizado para mudar o significado ou sentido de um sinal. Por exemplo, uma pessoa pode mudar sua expressão para indicar que está fazendo uma pergunta, usando o aumentativo ou o diminutivo de um substantivo.

2.1.2 Libras *versus* Língua Portuguesa

Não existe uma equivalência direta entre a Libras e a Língua Portuguesa; cada uma possui sua própria estrutura e são independentes entre si. Por exemplo, na Libras não existem artigos, preposições e conjunções, visto que os mesmos são incorporados em um sinal [BRITO, QUADROS e FELIPE, 2017].

Na Língua Portuguesa sentenças afirmativas, interrogativas, entre outras, são expressas através do uso de pontuações, como o sinal de exclamação ou a mudança da entonação da voz. No caso da Libras isso é realizado com a mudança da expressão facial ou pela incorporação de um movimento, como balançar a cabeça para negar um sinal. Também, na Libras não existe tempo verbal e, como consequência, a exposição de noções temporais é diferente, tornando necessário o acréscimo de um sinal que indica o tempo verbal (STROBEL e FERNANDES, 1998). Um exemplo de uma sentença comparando a estrutura gramatical das duas línguas é:

> Libras: FUTURO-EU-ESTUDAR-FACULDADE-COMPUTAÇÃO
> Português: Eu farei faculdade de computação

Nele, nota-se a ausência da preposição “de” e que os sinais FUTURO e ESTUDAR equivalem a palavra FAREI, demonstrando que pode não haver equivalência direta entre os sinais da Libras e palavras da língua portuguesa. Também, a frase em português segue a estrutura sujeito-verbo-objeto e, na Libras, essa estrutura não é comum.

Observando as características da Libras notam-se alguns dos desafios para desenvolver um sistema que traduza sinais da linguagem para o Português oral ou escrito. O primeiro desafio consiste em retornar uma palavra equivalente ao sinal representado em um vídeo de entrada, o que não é suficiente para estruturar uma frase com a mesma semântica. O segundo desafio consiste em organizar os sinais reconhecidos conferindo-lhes uma semântica equivalente entre as duas línguas e isso requer o entendimento do contexto em que eles se encontram, porém essa etapa não foi abordada neste trabalho. Outra característica da Libras não considerada foram as expressões faciais, para simplificar a base de dados a ser criada, pois julgou-se que essa característica aumentaria sua complexidade.

2.2 Redes Neurais Artificiais

Redes Neurais Artificiais (RNA) são uma subárea da inteligência artificial, mais especificamente do aprendizado de máquina. Elas representam um conjunto de modelos computacionais que têm como objetivo aproximar alguma função f, que mapeia uma dada entrada para uma respectiva saída, com uma outra função f'. No caso deste trabalho, queremos definir uma função que receba um vídeo como entrada e retorne uma categoria ou o sinal correspondente.

A unidade mais básica de uma rede neural artificial são os neurônios, exemplificado na Figura 2.2. Um neurônio pode receber uma ou várias entradas, que são ponderadas pelos pesos a elas associados; a combinação destas entradas e seus pesos produz um valor de saída para o neurônio, que é passado para uma função não-linear, como *Rectified Linear Unit* (ReLU), definido pela seguinte função $f(x) = max(0, x)$.

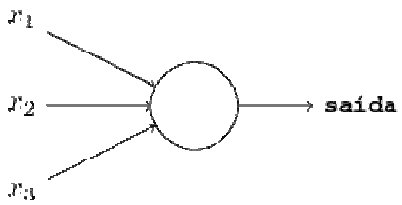

Figura 2.2: Um neurônio (componente básico de uma RNA) com três entradas
Fonte: Adaptado de NIELSEN (2015)

Os conjuntos de neurônios formam camadas, conforme ilustrado na Figura 2.3. Uma rede neural é composta por várias camadas que se dividem em três categorias. A primeira, a camada de entrada, é responsável por receber os dados que serão processados pela rede neural e não realiza nenhum tipo de transformação nos dados de entrada. Então, tem-se camadas escondidas, que serão responsáveis por realizar transformações lineares ou não-lineares sobre os dados de entrada. Por fim tem-se uma camada de saída, que retorna alguma informação útil, por exemplo, a probabilidade de um vídeo de entrada conter determinado sinal da Libras [GOODFELLOW, BENGIO e COURVILLE, 2016]. O número de camadas escondidas e o número de neurônios por camadas são variáveis e são parâmetros da arquitetura de uma RNA. Também, cada camada pode ter um número diferente de neurônios entre si.

Os neurônios podem ou não se conectarem com todos os outros neurônios da camada anterior. Quando o neurônio está conectado a um neurônio da camada anterior, esse usa a

saída deste como entrada, multiplicando-a pelo peso (valor numérico) associado à conexão entre eles. A soma associada ao produto das entradas com os pesos dos vários neurônios pode, ou não, ser transformada por uma função não-linear, gerando assim a saída da rede neural. Os pesos associados às entradas dos neurônios são os parâmetros de uma rede neural e neles reside o conhecimento da rede após realizada a fase de treinamento.

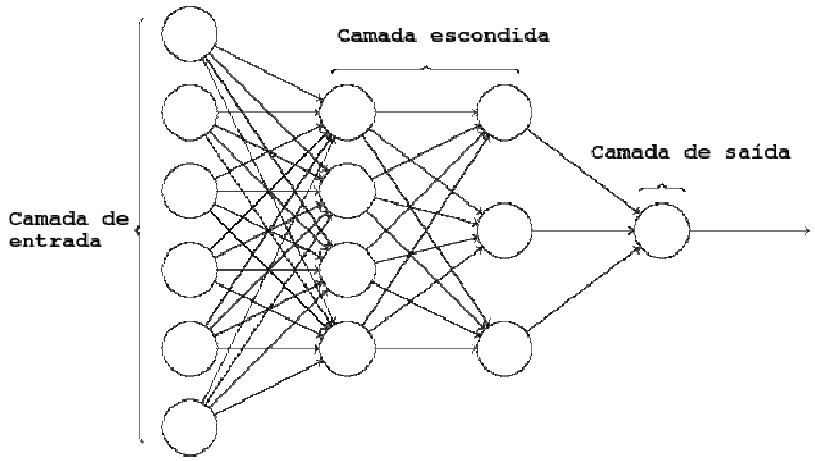

Figura 2.3: *Perceptron* de múltiplas camadas
Fonte: Adaptado de NIELSEN (2015)

Um dos modelos mais simples de rede neural é o *perceptron* de múltiplas camadas [GOODFELLOW, BENGIO e COURVILLE, 2016], representado na Figura 2.3. Ao ajustar os pesos dos neurônios, ou seja, ao mudar seus valores tem-se como resultado um valor f' que corresponde a uma melhor ou pior aproximação de f desejado. Esse processo de alterar os pesos das entradas dos neurônios da RNA, e consequentemente os coeficientes do modelo, é chamado de treinamento e para isso é necessária uma forma de dizer quão bem f' aproxima-se de f. Para isso uma função objetivo ou de perda é utilizada [LECUN, BENGIO e HINTON, 2015]. Se estiverem disponíveis exemplos de entrada e suas respectivas saídas pode-se calcular quão boa é a aproximação, comparando a saída esperada (f) com a saída retornada (f'). Normalmente os pesos são inicializados de forma aleatória, e diferentes estratégias podem ser empregadas.

O processo de ajustar os parâmetros do modelo utilizando dados conhecidos, quando a sua resposta esperada é conhecida, é chamado de aprendizado supervisionado [LECUN, BENGIO e HINTON, 2015]. Sendo o modelo de treinamento utilizado neste trabalho.

Existem várias arquiteturas de redes neurais que se diferenciam pela forma de como os neurônios se relacionam entre si. Este trabalho focou nas que são do tipo *feedforward*, isto é,

os neurônios só recebem dados como entradas daqueles que estão em camadas anteriores a eles. Um exemplo desse tipo de rede é o perceptron de várias camadas, ilustrado na Figura 2.3.

2.2.1 Aprendizagem e Treinamento

Treinar uma RNA significa achar o ponto que minimiza o valor retornado por uma função objetivo, logo ele se torna um problema de otimização computacional. Diferentes métodos podem ser utilizados, no entanto o mais empregado é conhecido como gradiente descendente em conjunto com o *backpropagation*. Usando cálculo de múltiplas variáveis, este método calcula os valores das derivadas parciais da função objetivo em relação a cada peso da RNA e os subtrai dos pesos atuais da rede. Esse processo é realizado por diversas iterações até a convergência para um mínimo local.

Para ilustrar esse conceito, considere a função $f(x) = x^2$, cujo mínimo da função é obtido quando $x = 0$. Se o valor de x for inicializado como 5, então ao aplicar este valor na derivada da função anterior ($f'(x) = 2x$) obtêm-se 10. Subtraindo este valor de x obteremos -5 e realizando novamente o processo, o valor de x retorna para 5. Esse comportamento é um dos problemas de métodos baseados no gradiente descendente pois, se os passos de atualização dos parâmetros da função forem muito grandes, a função pode não convergir a um mínimo.

Uma forma de contornar este problema é multiplicar o valor de $f'(x)$ por outra variável chamada de taxa de aprendizado. Para $x = 5$, tem-se que $f'(x) = 10$ e, com uma taxa de aprendizado de 10%, x será atualizado para x = 5 - 0,1.10 = 4. Nas próximas iterações x será atualizado para 3,2, depois para 2,56 e assim por diante até chegar a um valor próximo de 0, que equivale ao mínimo da função $f(x) = x^2$. O valor da taxa de aprendizado é um parâmetro a ser definido pelo usuário pois, dependendo do seu valor, a função pode não convergir ou pode demorar muito para isso.

2.2.2 Generalização e Overfitting

O teorema da aproximação universal diz que para uma função não-constante, com limites, monotônica contínua e positiva, existe uma RNA com apenas uma camada escondida e um número suficiente de neurônios capaz de representá-la, com um erro menor que ε [NIELSEN, 2015]. Por exemplo, no caso do aprendizado supervisionado, dado um conjunto finito de dados, uma RNA consegue achar uma função que mapeia os dados de entradas para seus

respectivos valores de saídas.

Porém, apesar disso não há garantia de que a RNA terá um bom desempenho em dados nunca vistos. O fenômeno no qual uma RNA tem bom desempenho em um conjunto de dados de treinamento, porém tem um desempenho ruim em dados nunca vistos, como um conjunto de dados de teste, é conhecido como *overfitting*. Isso ocorre devido ao modelo se ajustar perfeitamente aos dados de treinamento. Caso a RNA apresente desempenho equivalente para dados de treinamento e dados de teste, então diz-se que a RNA tem capacidade de generalização, o que é um comportamento desejável.

2.2.3 Redes Neurais Convolucionais

Redes neurais convolucionais (RNC) são um tipo especial de RNA para processar dados que estão organizados em formas de malhas, com uma ou mais dimensões [GOODFELLOW, BENGIO e COURVILLE, 2016]. Elas possuem esse nome pois aplicam uma operação conhecida como convolução sobre os dados de entrada. Uma convolução é composta por três elementos: os dados de entrada, um *kernel* ou filtro e os dados de saída ou vetor de características. O filtro, no universo 2D, corresponde a uma matriz com dimensões mxn de valores numéricos, chamados pesos. A Figura 2.4 exemplifica a operação de convolução sobre uma imagem. Nesta operação o filtro é posicionado sobre o pixel de coordenada (3, 3) da imagem e o produto entre os pesos do filtro e os pixels sobrepostos na imagem original são acumulados em um somador. O valor acumulado será atribuído à primeira posição da imagem resultado e, na sequência, o filtro será deslocado (convoluído) para o próximo pixel da imagem original, onde o tamanho desse deslocamento é um parâmetro configurável, chamado de *stride*. A operação de convolução estará concluída quando o filtro for convoluído sobre todos os pixels possíveis na imagem original. Os pesos do filtro são os parâmetros treinados pela RNC e vários filtros podem ser utilizados sobre uma mesma entrada.

Um dos problemas de utilizar uma rede neural, como o *perceptron* de múltiplas camadas, é que ela não considera a estrutura espacial de uma imagem. O uso de RNC facilita o treinamento e diminui o número de parâmetros necessários para problemas que envolvem o uso de imagens [NIELSEN, 2015].

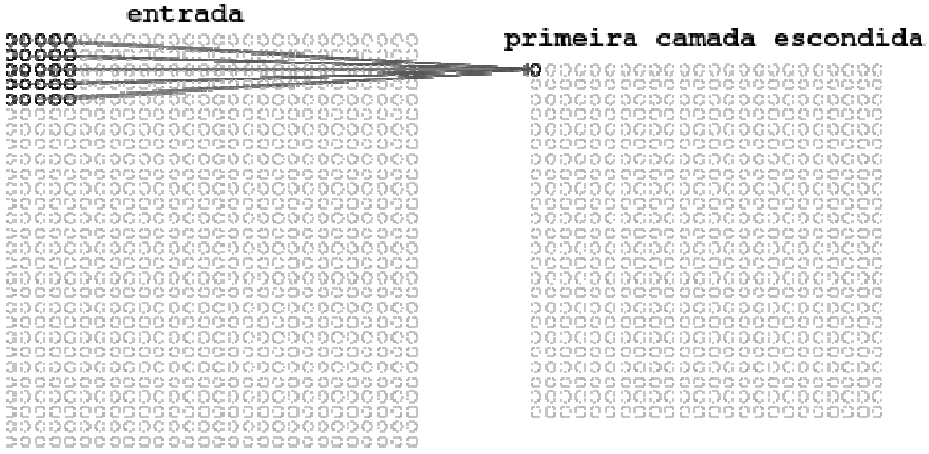

Figura 2.4: Processo de convolução sobre dados de entrada
Fonte: Adaptado de NIELSEN (2015)

As camadas iniciais das RNC, correspondentes aos filtros de convolução, são responsáveis por detectar características, como arestas em uma determinada região da imagem. Assim, várias camadas podem ser utilizadas em sequência para detectar cada vez mais características de alto nível, como a presença de olhos em imagens que contenham pessoas, por exemplo. É através do uso de diversos filtros, onde cada um é treinado para detectar uma característica diferente, que uma RNC consegue diferenciar a classe de diferentes imagens. A característica detectada por cada um dos filtros é determinada automaticamente durante a fase de treinamento, em que é responsabilidade da própria rede descobrir o conjunto de características mais interessantes para diferenciar e classificar de forma apropriada as diferentes instâncias de entrada.

Para ilustrar essa ideia de como um filtro detecta uma determinada característica, considere a Figura 2.5, que corresponde aos valores numéricos de um filtro de convolução à esquerda e a característica que ele detecta à direita. Note que os valores diferentes de zero no filtro formam uma curva semelhante à característica desejada. Com isso, ao realizar a operação de convolução, dados de entradas que possuem uma forma semelhante à característica que o filtro detecta, resultará em valores numéricos grandes. Para dados de entradas que possuam formas muito diferentes os valores apresentados serão baixos.

Como o filtro é convoluído por toda a imagem, caso a característica representada pelo filtro esteja presente, ela será detectada e, posteriormente, essa informação será utilizada por outras camadas de mais alto nível, para hierarquicamente detectar características de mais alto nível. Porém, note que se essa mesma característica estiver rotacionada, por exemplo, o filtro não a detectará. Nesse sentido é interessante aumentar artificialmente os dados de entrada,

realizando operações geométricas afins, como rotação, para tornar o reconhecimento da RNC mais robusto.

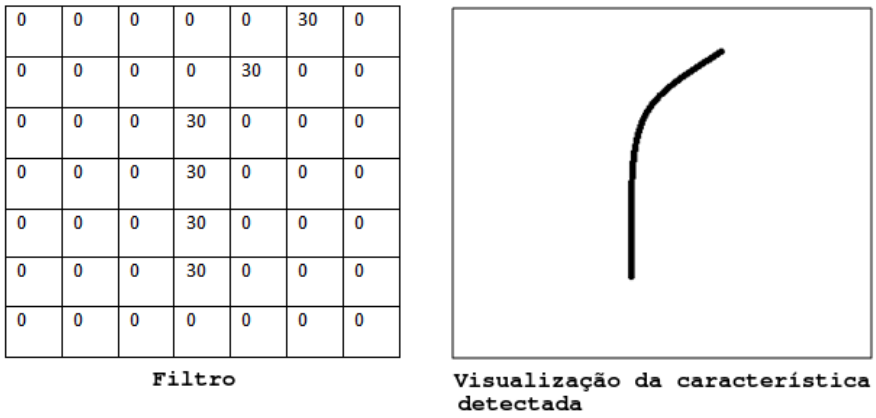

Figura 2.5: Ilustração de um filtro detector de característica
Fonte: Adaptado de DESHPANDE (2016)

2.3 Trabalhos Correlatos

Este trabalho não é o primeiro a tentar reconhecer de forma automática uma língua de sinais. Neiva e Zanchettin (2018) publicaram um *survey* abrangendo 43 trabalhos, de 2009 até 2017, com diferentes tipos de dados de entrada e técnicas de reconhecimento, focando no contexto de dispositivos móveis. Suharjito *et al*. (2017) também apresentam um *survey* sobre o assunto, com foco nos procedimentos de aquisição da base de dados utilizadas. Nesta seção serão apresentados trabalhos de outros autores, apontando algumas diferenças quando comparados a este.

2.3.1 Métodos de aquisição da base de dados

Diferentes equipamentos foram utilizados para adquirir os dados a serem utilizados no treinamento dos métodos de reconhecimento. Entre eles têm-se *webcams* [ZHANG, 2004], como as que foram utilizadas neste trabalho. Câmeras especiais como o *Microsoft Kinnect* também foram utilizados [WANG, 2015]; além das imagens em cor, elas também registram a profundidade dos elementos presentes na cena, permitindo distinguir o fundo de um indivíduo de uma forma fácil e rápida. *Hardwares* especiais como luvas [GAO, FANG e CHEN, 2004] e acelerômetros [LOKHANDE, 2015] são outros equipamentos que já foram utilizados. Nestes, o reconhecimento é feito utilizando sinais digitais produzidos por tais equipamentos e

não imagens.

A principal vantagem de utilizar uma *webcam* ou uma câmera comum é que elas podem ser encontradas com um preço acessível, já câmaras especializadas como o *Kinnect* são mais caras. Essa característica do preço também é válida para *hardwares* especiais. Além disso, esse tipo de equipamento pode não ser conveniente para o usuário, pois seu uso pode ser desconfortável ou incômodo.

2.3.2 Bases de dados

As bases de dados utilizadas podem ser categorizadas em três: aquelas que possuem apenas sinais estáticos, aquelas que possuem sinais dinâmicos e aquelas que possuem ambas. Outra característica é se a base de dados consiste em imagens, vídeos ou outro tipo de dado e, no caso de vídeos, se eles foram gravados em ambientes controlados ou não. Além disso, pesquisadores já trabalharam com várias Línguas de Sinais, como a *American Sign Language* e *Chinese Sign Language*, como mostram os *surveys*.

Em específico na Libras, Teodoro (2015) construiu uma base de dados contendo 30 sinais realizados por dez diferentes indivíduos e coletados por câmera em ambientes não controlados e com fundo não estático. No total 20 vídeos foram gravados, dois com cada indivíduo e cada um contém todos os sinais, porém esses vídeos foram segmentados em 30 vídeos menores, cada um contendo um sinal. A resolução utilizada na gravação foi 640x320 e os sinais selecionados são dinâmicos. Monteiro (2016) também construiu sua própria base de dados, contendo 548 sequências de vídeos, cada um contendo 24 sinais e vinte indivíduos. O ambiente utilizado para gravação consiste em fundos estáticos e dinâmicos. Apesar de não estar claro, subentende-se que nestes trabalhos um único ângulo de visão foi considerado, sendo ele frontal e reto.

2.3.3 Métodos de reconhecimento

Diversos métodos e metodologias já foram experimentadas para reconhecer sinais e gestos de uma forma geral. A tarefa de reconhecimento de um sinal ou gesto pode ser dividida em duas etapas. A primeira é extrair as características da entrada e a segunda é utilizar essas características para o reconhecimento. Neiva e Zanchettin (2018) revisaram 22 diferentes técnicas diferentes de extração de características. Segundo eles, técnicas baseadas em segmentação de pele são bem populares. No caso do reconhecimento, segundo os mesmos autores, métodos como *Support Vector Machine*, *K-Nearest Neighbors*, redes neurais, entre

outros, foram utilizados para sinais estáticos. Já no caso de sinais dinâmicos tem-se os métodos *Hierarchichal Temporal Memory*, *Dynamic Time Warping*, redes neurais, entre outros.

2.3.4 Considerações finais

Diferente de outros trabalhos levantados até o momento, este trabalho considerou apenas fundo estático e uniforme na construção da base de dados, porém utilizaram-se câmeras de vários modelos posicionadas em diferentes ângulos, ampliando o número de instâncias de um mesmo sinal da Libras na base de dados. Além disso, o número de amostras coletadas, o número de sinais considerados e o número de voluntários utilizados é maior.

Um problema que dificulta a comparação entre diferentes trabalhos, inclusive com este, é que são empregadas diferentes bases de dados e métodos de reconhecimento. Além disso, a maioria das bases de dados não estão disponíveis publicamente, são de difícil acesso e são construídas considerando a Língua de Sinais de sua região, motivando outros pesquisadores a construir sua própria base de dados. Ademais, as bases de dados utilizadas, mesmo aquelas compostas por milhares de sinais, em sua maioria apresentam poucas amostras de um mesmo sinal. Por exemplo, alguns trabalhos revisados por Suharjito *et al*. (2017) possuem uma amostra por sinal igual a 5, 10 e 28. Mais informações podem ser encontradas nos *surveys* citados.

O pequeno tamanho das bases de dados, nos trabalhos revisados nos dois *surveys*, leva os métodos de reconhecimento a obterem uma acurácia[1] média de aproximadamente 87% com um desvio padrão de aproximadamente 12%. Porém, pouco se pode afirmar sobre o desempenho do método em situações reais, que envolvam diferentes pessoas e sinais.

1 . Índice percentual correspondente ao número de acertos de um classificador em relação ao total de amostras analisadas.

Capítulo 3
Metodologia

Para atingir os objetivos deste trabalho foi necessária uma ampla base de dados para treinamento da rede neural. Por esse motivo optou-se por criar conjunto de dados próprio, que está disponível de forma pública[2], mediante o preenchimento de um formulário com informações básicas, como o nome, instituição e quais os interesses e usos da base. Por fim é necessário concordar com os termos de uso da base de dados.

Para compor conjunto foram gravados vídeos de voluntários realizando sinais da Libras. Entre os voluntários estão docentes e discentes da Unioeste. Os anexos A e B apresentam o termo de consentimento livre e esclarecido (TCLE) e o parecer do Comitê de Ética da universidade a respeito do projeto, respectivamente.

As próximas seções apresentaram os procedimentos e equipamentos utilizados, bem como as considerações relevantes para preparação e montagem do ambiente para captura dos vídeos. Também serão apresentadas quais ferramentas foram utilizadas e quais ações foram tomadas para implementar, treinar e testar a arquitetura de RNC usada neste trabalho.

3.1 Construção da base de dados

Os voluntários foram organizados em grupos de 16 indivíduos e cada grupo realiza um conjunto de 12 sinais da Libras, selecionados com ajuda de uma intérprete. Para gravar os vídeos foram utilizadas 9 *webcams*, de 8 diferentes marcas e modelos, com qualidade de vídeo variada. Algumas apresentam uma alta taxa de ruído e/ou baixa fluidez das imagens se comparadas às câmeras de maior qualidade, presentes no conjunto.

A Figura 3.1 mostra um exemplo de um quadro de cada câmera, em que o indivíduo presente na foto está em repouso. Nela pode-se observar as diferenças de qualidade e posicionamento das câmeras empregadas na captura dos vídeos.

2 . https://goo.gl/forms/bQKc1TFVTukTBRY92

Figura 3.1: As diferentes imagens de um indivíduo em repouso capturadas pelas câmeras utilizadas na criação da base dados

A Figura 3.2 mostra outro exemplo com o indivíduo movimentando rapidamente seu braço direito. Pode-se observar neste exemplo como a fluidez das câmeras é diferente durante a captura de um vídeo. No primeiro quadro o braço praticamente desaparece. Contudo, é importante ressaltar que o movimento para realizar um sinal, em geral, não tem a mesma velocidade empregada nesse exemplo.

Figura 3.2: As diferentes imagens de um indivíduo em movimento capturadas pelas câmeras utilizadas na criação da base dados

A Tabela 3.1 mostra informações das webcams utilizadas na captura dos vídeos. Os nomes das câmeras são apresentados pela identificação realizada pelo sistema operacional Linux, obtidos por meio do comando *lsusb*. O USB ID representa um identificador único para um dispositivo USB segundo a estrutura "Código do Fabricante:Dispositivo". O *Pixel Format*, em poucas palavras, diz respeito à codificação usada na transmissão dos *frames* (imagens) dos vídeos capturados pela câmera para o dispositivo de processamento.

Tabela 3.1 - Informações sobre as câmeras

Nome	USB ID	Resolução Utilizada	Quantidade	Pixel Format
Creative Labs Technology, Ltd USB Webcam NX [PD1110]	041e:401c	352x288	1	JPEG
Cubeternet Webcam	1e4e:0100	640x480	1	JPEG
Cubeternet GL-UPC822 UVC Webcam	1e4e:0102	640x480	1	YUYV
KYE Systems Corp. (Mouse Systems) VideoCAM Web	0458:700f	640x480	1	JPEG
Logitech, Inc. Webcam C6000	046d:0808	640x480	1	YUYV
Microdia PC Camera with Mic (SN9C105)	0c45:60fc	640x480	1	JPEG
Microsoft Corp. LifeCam HD-5000	045e:076d	640x480	2	JPEG
Z-Star Microelectronics Corp. ZC0301 Webcam	0ac8:301b	640x480	1	JPEG

Os vídeos foram gravados com a resolução 640x480, com exceção de uma câmera, que não suporta essa resolução, como mostra a Tabela 3.1. Elas foram montadas em um suporte, do tipo tripé, posicionadas em uma altura máxima de 1,80 metros e organizadas de tal maneira que cada uma capturava os vídeos em um ângulo diferente, formando uma configuração de matriz. Das 9 câmeras, 3 foram posicionadas em frente aos voluntários em 3 diferentes ângulos verticais. A primeira delas apontada para frente, paralela ao piso, a segunda apontada ligeiramente para baixo e a terceira apontada ligeiramente para cima. As outras 6 câmeras apontam levemente para a lateral do voluntário, 3 em cada lado, com um ângulo variando entre 7,5° e 20°, com apontamento vertical análogo ao usado nas câmeras frontais. A Figura

3.3 mostra a estrutura utilizada na aquisição dos vídeos exemplificando o que foi descrito.

Figura 3.3: Estrutura utilizada na aquisição dos vídeos

O ambiente utilizado para gravar os vídeos possui uma iluminação não uniforme que variava de acordo com o período do dia. O fundo foi formado por um tecido verde e as lâmpadas presentes foram ligadas e desligadas de forma aleatória com alguns voluntários visando diversificar as condições de iluminação presente nos vídeos que comporão a base de dados. Foi utilizado um único computador para processar e salvar os vídeos em disco. Assim, as câmeras gravam os vídeos de forma sincronizada. As configurações das câmeras utilizadas seguem os padrões do fabricante, com exceção dos modelos da Microsoft que, em algumas situações teve seu foco automático desativado e definido manualmente, e do modelo da *Creative Labs* que teve o valor *gamma* alterado para corrigir o brilho das imagens.

Cada vídeo registrado na base contém um voluntário sinalizando um único sinal da Libras, com duração entre 1 e 2 segundos, a aproximadamente 30 quadros por segundo. O voluntário repetiu um mesmo sinal 5 vezes com variações sutis entre elas, como a altura da mão, amplitude do movimento das mãos ou velocidade de realização do sinal, por exemplo. Dessa forma, para cada voluntário foram coletados 540 vídeos, resultando em 8640 vídeos por grupo e 720 vídeos por sinal. A configuração do computador empregado na construção da base é descrita a seguir.

- Elementary OS 0.4.1 Loki 64-bits, Linux 4.13.0-38-generic;
- Dual-Core Intel i3-7100 CPU @ 3.9 GHz rev04;
- Placa Mãe H110M-HG4 ASRock;
- 8 GB RAM;
- Controlador PCI-e USB 3.0 5 Gbps, 4 portas.

No total, 4 câmeras foram conectadas no controlador PCI-e e as outras cinco diretamente na placa mãe do computador. Um *hub* USB de quatro portas foi utilizado para conectar o teclado e mouse no computador.

3.2 Ferramentas utilizadas

Para auxiliar na avaliação do desempenho de uma arquitetura de RNC na tarefa de reconhecimento de sinais da Libras, algumas ferramentas foram utilizadas.

A primeira delas foi a *Application Programming Interface (*API) Keras [CHOLLET, 2015], que fornece uma interface de alto nível para RNA e permite uma fácil e rápida prototipagem, suportando diferentes tipos e arquiteturas. As redes neurais construídas com esta API são executadas, de forma transparente, utilizando processamento da CPU e GPU. Além disso, ela fornece um *front-end* para outros *frameworks* como o TensorFlow, uma biblioteca de computação numérica baseada em fluxo de dados em grafos [ABADI, 2015], que é utilizada neste trabalho. A biblioteca OpenCV [OpenCV Team, 2018] foi empregada na programação do sistema de coleta e tratamento dos vídeos. A linguagem de programação Python 3 [Python Core Team, 2008] foi empregada na programação do sistema de captura de vídeos, bem como na programação das redes neurais desenvolvidas para analisar os vídeos e reconhecer os sinais da Libras.

3.3 Pré-processamento

Os vídeos que constituem a base sofreram algumas operações de pré-processamento. Inicialmente reamostrou-se os vídeos para uma resolução de 320x240 pixels e, em seguida, recortou-se para uma janela com 224x224 pixels, posicionada arbitrariamente, toda vez que um vídeo foi apresentado para a rede neural durante o treinamento. A justificativa para reduzir a resolução é aumentar a velocidade do treinamento da RNC.

Após a reamostragem e o recorte utilizou-se duas transformações geométricas: rotação e cisalhamento (sobre os eixos x e y de cada imagem dos vídeos); em ambas as operações considerou-se um ângulo aproximadamente 15°. Além dessas duas transformações, rotacionou-se arbitrariamente os vídeos em 180° ao redor do eixo y, com o objetivo de aumentar artificialmente a diversidade do conjunto de dados. Na sequência, selecionou-se um subconjunto dos *frames* de cada vídeo, escolhendo-se aleatoriamente um entre os quatro

frames iniciais, e então se salta de dois em dois até atingir um total de 12 *frames*.

Os vídeos foram divididos em três conjuntos: treinamento, validação e teste. O conjunto de treinamento possui 75% dos dados, já os outros dois possuem 12.5% cada, que após divididos não foram alterados. Os vídeos do primeiro conjunto são aqueles que a rede neural utiliza durante seu aprendizado. Os do segundo conjunto servem para otimizar a arquitetura e os parâmetros da rede neural; após treinada a rede, sua acurácia é verificada avaliando os vídeos deste conjunto. Definidos os melhores parâmetros, a rede foi treinada com a união desses dois conjuntos e então se mede a acurácia final com o conjunto de teste. Essa divisão foi feita de duas formas, a primeira é não compartilhando vídeos de um mesmo voluntário entre os conjuntos e a outra compartilhando.

3.4 Implementação

Diversas arquiteturas de RNC foram testadas em experimentos preliminares e, a partir destes experimentos empíricos, optou-se pela arquitetura de RNC que apresentou os melhores resultados. A arquitetura selecionada é composta por três componentes: o primeiro é uma rede pré-treinada para extrair características espaciais dos vídeos. O segundo é uma rede responsável por extrair as características espaço-temporais e o terceiro componente é uma rede que utiliza as características extraídas pelas redes anteriores para identificar qual sinal da Libras está presente em um vídeo.

O primeiro componente da arquitetura de RNC selecionada é conhecida como MobileNetV2 e encontra-se disponível na API Keras. O uso do MobileNetV2 justifica-se por ser uma arquitetura projetada para ser executada em plataformas como celulares, que possuem poucos recursos computacionais se comparado a computadores de mesa, e é uma plataforma interessante para aplicações envolvendo tradução automática de Libras para português, devido ao seu bom desempenho, que é comparável a outras arquiteturas como InceptionV3 e ResNet50, utilizando menos recursos e parâmetros [SANDLER, 2018]. A Tabela 3.2 mostra um comparativo entre essas arquiteturas, em que a acurácia foi avaliada com o conjunto de dados ImageNet. A coluna *top-1* diz respeito à porcentagem de acerto da arquitetura em predizer corretamente a classe de uma determinada imagem e o *top-5* é quando a classe esperada está entre aquelas que a arquitetura acredita que sejam as cinco mais prováveis. Mais detalhes do funcionamento da MobileNetV2 podem ser encontrados em [SANDLER, 2018].

Tabela 3.2: Comparação de características e acurácia de 3 arquiteturas de RNC quando aplicadas na classificação de imagens da base ImageNet

Arquitetura	Tamanho em memória	Top-1 (Acurácia)	Top-5 (Acurácia)	Número de parâmetros	Número de camadas
MobileNetV2	17 MB	0.665	0.871	4.253.864	88
InceptionV3	92 MB	0.788	0.944	23.851.784	159
ResNet50	99 MB	0.759	0.929	25.636.712	168

Fonte: CHOLLET, 2015

O segundo componente é composto por uma variação de vários módulos concatenados em sequência, conhecidos como *spatiotemporal-separable 3D convolutions* (S3D) descrito em XIE (2017). Ele é composto por várias operações de convolução, com filtros 3D, de diferentes tamanhos operando sobre os dados de entrada. Na sequência, os resultados das operações são concatenados e operados por um próximo filtro (camada). A razão para este procedimento é deixar o próprio treinamento descobrir qual o melhor filtro para um determinado tipo de entrada. Um filtro 3D funciona de forma análoga a um 2D, porém operando sobre dados com três dimensões como um vídeo. O Apêndice A apresenta segmentos dos códigos que o implementa.

Por fim, para o terceiro componente tem-se três camadas totalmente conectadas, constituindo um *perceptron* de múltiplas camadas. Apenas o segundo e o terceiro componentes foram treinados, pois a MobileNetV2 tem a opção de utilizar os pesos pré-treinados com o conjunto de imagens da base ImageNet. Além disso, testes mostraram que utilizá-la com pesos iniciados de forma arbitrária resulta em um pior desempenho da RNC. Para treinar a rede, utilizou-se uma máquina com uma CPU AMD FX-6300 com 12 GB de memória RAM e uma GPU GTX 760 de 2GB.

3.5 Treinamento

O treinamento de uma RNC consiste em apresentar para ela todo o conjunto, definindo uma época. Diversas épocas podem ser utilizadas até que a RNC pare de convergir, isto é, pare de melhorar seu desempenho, a partir de uma determinada métrica, neste caso chamada de *categorical crossentropy*, implementada pela API Keras. Neste trabalho treinou-se cada arquitetura de RNC durante 10 épocas. A justificativa é que utilizar mais épocas não mostrou nenhuma melhora significativa.

Para cada época utilizou-se um *batch* de tamanho 4, isto é, apresentou-se para a RNC 4 vídeos por vez e os erros de classificação dessas instâncias foram utilizadas em conjunto para calcular o gradiente para atualizar os pesos da RNC. Valores de *batchs* maiores resultam em um tempo de treinamento menor e seu valor pode determinar quão bem a RNC converge e sua acurácia final. Neste trabalho limitou-se a um *batch* de tamanho 4 devido à limitação de memória da placa de vídeo utilizada.

Utilizou-se como taxa de aprendizado inicial o valor 0.05, pois, de acordo com os experimentos realizados, este valor apresentou os melhores resultados, além de serem mais estáveis. Isto é, valores maiores diversas vezes resultaram na não convergência da RNC e valores menores apresentaram resultados inferiores, utilizando a arquitetura apresentada anteriormente.

Além disso, para cada época reduziu-se a taxa de aprendizado para 90% de seu valor atual e, caso a acurácia no conjunto de validação não apresente melhora, a cada duas épocas divide-se a taxa de aprendizado por 10. Essa operação visa aumentar a estabilidade da convergência à medida que se realiza o treinamento da rede.

3.6 Métricas de avaliação de desempenho

Para analisar o desempenho das RNC adotaram-se duas formas de avaliação, além da análise da acurácia do conjunto de treinamento e teste. A primeiro é a matriz de confusão e a segunda é o *Top-N*.

3.6.1 Matriz de confusão

A matriz de confusão é uma tabela que permite visualizar os acertos e erros quando instâncias são apresentadas a um classificador. A Tabela 3.3 representa um exemplo hipotético de sistema de classificação de três classes (**A**, **B** e **C**) com 15 instâncias cada. Cada linha representa a classe esperada e cada coluna a classe predita. Por exemplo, a classe **A** possui 12 instâncias preditas de forma correta e três de forma incorreta, preditas como classe **B**. De forma análoga a classe **B**, possui 10 instâncias classificadas corretamente e 5 classificadas incorretamente, preditas como classe **A**.

Uma informação que a matriz de confusão permite analisar é onde as predições incorretas estão, o que pode auxiliar na compreensão do processo de classificação. Isto é, poderíamos concluir que as classes **A** e **B** possuem características semelhantes e ambas são

significativamente diferentes da classe **C**, pois para as instâncias pertencentes a essas classes, nenhuma foi classificada como **C**.

Tabela 3.3: Exemplo de uma matriz de confusão

		Classe predita		
		A	B	C
Classe esperada	A	**12**	3	0
	B	5	**10**	0
	C	1	0	**14**

3.6.2 *Top-N*

Alguns classificadores tem como saída um vetor de probabilidade, em que uma dada posição representa a probabilidade de uma instância ser de uma determinada classe. Aquela que possui a maior probabilidade de corresponder a uma determinada instância é a classe predita pelo classificador. Em alguns casos é possível que, ao errar a classe correta, esta se encontre entre as mais prováveis. Problemas como a tradução de uma linguagem para outra pode se utilizar do contexto para decidir qual é a palavra/sinal correta em determinado contexto, dentro de uma sentença. Nesse sentido, medir a acurácia, em termos de acerto ou não da classe, pode não ser a forma mais adequada de avaliar o desempenho de um classificador.

O *top-N* considera uma predição correta se a classe esperada está entre as *N* mais prováveis segundo o classificador.

Capítulo 4
Resultados e Discussões

4.1 Base de dados

Em relação ao primeiro objetivo deste trabalho construiu-se uma base de dados com quatro grupos compostos por doze sinais. No total, 65 voluntários tiveram seus gestos capturados e 48 sinais considerados, totalizando aproximadamente 34.560 vídeos para serem disponibilizados em uma base de dados aberta.

Os sinais escolhidos buscaram atender um de dois critérios: possuir pares de sinais que são semelhantes entre si e serem comuns no uso do dia-dia. A Tabela 4.1 mostra os sinais presentes na base de dados separados por seus respectivos grupos.

Tabela 4.1: Sinais presentes na base de dados

Grupo 1	**Grupo 2**	**Grupo 3**	**Grupo 4**
agosto	bom	assistir	como
avisar	casa	ele/ela	porque
me-avisar	dia	eu	sim
branco	estuda	família	não
educado	idade	gostar	dúvida
entender	local	ler	tchau
não-entender	noite	meu	vontade
esquecer	meu-nome	nós	ainda
pessoa	seu-nome	passear	faculdade
quente	oi	ter	futuro
rápido	tarde	tv	hoje
sentimento	trabalha	você	passado

Os sinais considerados semelhantes são aqueles que entre si mudam um ou poucos parâmetros que os definem. Por exemplo, o sinal meu-nome e seu-nome diferem unicamente na orientação da mão em relação ao indivíduo, conforme ilustrado na Figura 4.1. Note que a

única diferença é para onde a mão está orientada, sendo que o sinal meu-nome tem a palma da mão direcionada para quem sinaliza e o sinal seu-nome a palma está direcionada para quem o sinal é destinado. Na Tabela 4.2 os sinais que tem alguma semelhança estão agrupados.

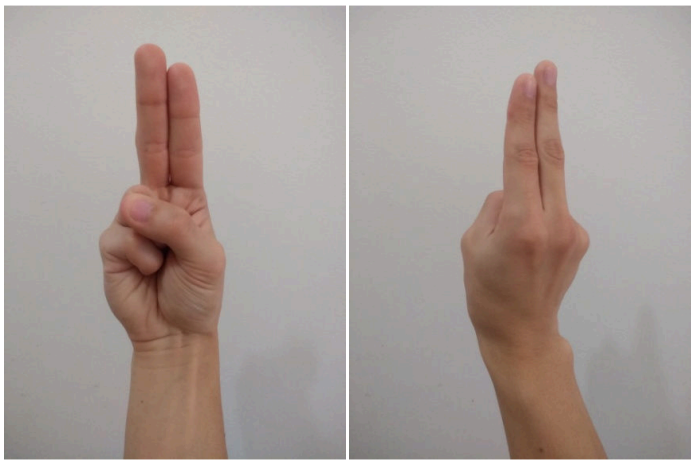

Figura 4.1: Diferença entre o sinal “seu-nome” (esquerda) com “meu-nome” (direita)

Tabela 4.2: Sinais presentes na base de dados e que se assemelham

agosto, gostar, meu, sentimento	dúvida, vontade	quente, rápido
avisar, me-avisar, futuro	entender, não-entender	meu-nome, seu-nome
branco, educado	esquecer, pessoa	ele/ela, você

4.2 Experimentos

Quanto ao segundo objetivo deste trabalho, avaliou-se a arquitetura escolhida classificando cada grupo de sinais usando duas estratégias de divisão do conjunto de dados. Para cada grupo os vídeos foram separados em conjuntos de treinamento, validação e teste seguindo estas duas estratégias: na primeira, utilizou-se os vídeos de um voluntário em apenas um dos conjuntos (treinamento, validação ou teste) e, na segunda, atribuíram-se aleatoriamente os vídeos em cada conjunto, mantendo a proporção dos sinais em cada um deles.

A Tabela 4.3 mostra os resultados obtidos de acordo com a primeira estratégia descrita anteriormente. Nela são apresentadas a acurácia em termos das métricas *top-1*, *top-3* e *top-5* sobre o conjunto de teste, a acurácia do conjunto de treinamento (A.C.T.), considerando predição correta apenas quando a RNC acertou a classe da instância do conjunto de treinamento e, por último, o tempo de treinamento total (T.T.) em segundos. O valor da acurácia varia entre 0 e 1.

Durante os experimentos realizados notou-se uma instabilidade nos resultados obtidos após a RNC ser treinada. Os valores apresentados na Tabela 4.3 foram os mais comuns após a RNC ser treinada várias vezes. Em alguns momentos a rede não convergiu, isto é, apresentou um desempenho semelhante a um método randômico e, em outras situações, apresentou um desempenho melhor do que os apresentados, com acurácia para o *top-1* ultrapassando valores de 0.60 para alguns grupos de sinais.

Tabela 4.3: Índices de acurácia para reconhecimento dos sinais em vídeos usando a primeira estratégia de divisão dos conjuntos

Grupos	*Top-1*	*Top-3*	*Top-5*	A.C.T.	T.T. [s]
Grupo 1	0.438	0.827	0.933	0.697	6627
Grupo 2	0.458	0.778	0.902	0.793	7429
Grupo 3	0.428	0.741	0.869	0.776	7410
Grupo 4	0.314	0.649	0.821	0.683	7231
Todos	0.190	0.383	0.504	0.733	27842

Comparando-se a acurácia do conjunto de treinamento e teste, existe uma diferença significativa entre eles, ou seja, houve *overfitting* em todos os grupos avaliados. Os gráficos apresentados no Apêndice B, que comparam a acurácia obtida com o conjunto de treinamento e validação, mostram claramente que há *overfitting*, pois a distância entre as duas curvas aumenta no decorrer do treinamento. Tentou-se realizar algumas ações visando melhorar esse resultado, como diminuir o número de filtros em cada camada convolucional da arquitetura, porém essa ação apenas resultou em desempenho inferior, não resolvendo o problema. Em outra ação aumentaram-se os valores máximos dos ângulos das operações de cisalhamento e rotação e, no caso do grupo 4, usaram-se os vídeos do voluntário que não concordou em ceder sua imagem, porém em nenhuma destas abordagens obteve-se melhora significativa.

Analisando-se os valores da acurácia para o *top-3* percebe-se o incremento na acurácia que, no caso do grupo 4, aumentou em aproximadamente 161%. Nos outros grupos notou-se também melhora parecida. A acurácia para o *top-5* também apresentou melhora, porém em menor proporção. Esses resultados tornam-se ainda mais interessantes pois, em etapa posterior à classificação, pode-se selecionar um dos sinais mais prováveis em função do contexto da sentença em análise.

Porém é importante destacar que estar entre os 3 e 5 mais prováveis, consiste em considerar os 25% e 41% de todas as classes, respectivamente, já que um grupo de sinal possui 12 sinais.

Quando considerados todos os 48 sinais, observa-se que a arquitetura de RNC desenvolvida apresentou valores baixos para a acurácia, chegando a ser mais que duas vezes menor do que os valores obtidos na análise de grupos de sinais, e bem menor se comparado a acurácia obtida com seu conjunto de treinamento. Porém, ao se considerar o nível de dificuldade, nos grupos individuais a probabilidade de se classificar um sinal corretamente e ao acaso é de 8% e, quando agrupados todos os sinais, esta probabilidade diminui para 2%. Portanto, em grupos individuais, o classificador possui um desempenho 5,4 vezes superior ao acaso, aumentando para 9,5 vezes quando agrupados todos os sinais. Com isso, podemos ver que apesar da acurácia ser menor, o classificador conseguiu um desempenho superior quando submetido a uma maior quantidade de dados.

As Tabelas 4.4, 4.5, 4.6 e 4.7 mostram as matrizes de confusão resultantes de cada grupo de sinal. No Apêndice C apresentam-se tabelas da matriz de confusão quando se analisaram todos os 48 sinais. As células das tabelas em cinza mostram para cada classe quantas instâncias foram classificadas corretamente. As células em laranja mostram sinais que possuem características semelhantes.

Destaca-se a Tabela 4.4, que contém a matriz de confusão para o grupo 1 que possui pares de sinais semelhantes entre si; na maioria dos casos, com exceção de dois, o segundo sinal mais classificado de uma determinada classe é aquela do sinal que possui características semelhantes. Esse resultado é interessante, pois fornece um indício de que a RNC está aprendendo a analisar os vídeos a partir de características de mais alto nível, e não apenas memorizando os padrões presentes neles, para uma determinada classe.

Tabela 4.4: Matriz de confusão (Grupo 1)

Sinais	agosto	avisar	avisar-me	branco	educado	entender	entender-não	esquecer	pessoa	quente	rápido	sentimento
agosto	64	0	6	2	1	0	0	0	0	1	2	13
avisar	11	28	20	3	7	1	1	0	0	8	10	1
avisar-me	17	4	37	2	6	0	0	0	0	8	2	5
branco	1	0	4	71	13	0	0	1	0	1	0	1
educado	0	0	0	38	51	0	0	0	0	0	0	1
entender	8	1	1	1	11	50	15	0	1	0	0	2
entender-não	1	0	0	1	8	24	45	5	1	0	2	2
esquecer	0	0	0	1	3	0	4	68	14	0	0	0
pessoa	0	2	2	6	0	0	1	40	37	0	1	0
quente	13	1	4	13	7	0	0	0	0	45	0	2
rápido	27	1	13	6	3	0	0	0	0	13	18	4
sentimento	39	1	5	5	5	0	0	0	0	2	0	33

Tabela 4.5: Matriz de confusão (Grupo 2)

Sinais	bom	casa	dia	estuda	idade	local	noite	nome-meu	nome-seu	oi	tarde	trabalha
bom	47	0	9	12	4	0	6	1	1	3	5	2
casa	2	45	1	32	0	1	6	0	2	0	0	1
dia	11	4	53	2	3	0	1	0	0	7	8	1
estuda	0	2	0	68	1	1	16	0	0	0	1	1
idade	0	0	7	3	55	0	6	0	1	2	12	4
local	1	1	0	20	0	39	12	4	3	2	1	7
noite	1	6	0	32	2	2	40	0	0	0	6	1
nome-meu	4	0	1	8	0	6	0	30	22	13	3	3
nome-seu	5	0	1	14	3	4	3	20	21	15	3	1
oi	9	0	2	6	7	1	6	11	14	27	4	3
tarde	16	0	4	12	0	1	6	2	2	4	43	0
trabalha	3	16	0	22	1	1	15	0	0	0	1	31

Tabela 4.6: Matriz de confusão (Grupo 3)

Sinais	assistir	ele-ela	eu	família	gostar	ler	meu	nos	passear	ter	tv	você
assistir	66	3	3	0	0	2	0	1	0	13	0	2
ele-ela	7	23	2	4	6	6	4	8	0	7	0	23
eu	3	3	27	0	2	8	19	5	0	14	0	9
família	1	5	2	55	4	13	0	2	0	3	2	3
gostar	0	1	1	5	28	13	22	1	1	15	1	2
ler	2	1	0	4	0	67	1	0	7	7	1	0
meu	0	1	8	0	1	18	46	3	1	12	0	0
nos	10	8	6	1	4	5	11	37	1	3	0	4
passear	6	1	1	4	0	31	0	0	42	5	0	0
ter	1	4	7	0	3	8	27	2	0	37	0	1
tv	2	1	0	13	0	46	0	1	3	1	22	1
você	7	23	4	1	0	7	5	9	1	18	0	15

Tabela 4.7: Matriz de confusão (Grupo 4)

Sinais	ainda	como	dúvida	faculdade	futuro	hoje	não	passado	porque	sim	tchau	vontade
ainda	23	13	0	3	6	4	10	0	0	2	4	25
como	6	9	6	1	0	6	11	1	0	2	2	1
dúvida	0	0	51	1	0	0	0	5	1	0	0	31
faculdade	11	1	0	62	9	0	1	1	0	1	0	4
futuro	13	1	0	1	37	3	13	2	0	1	8	11
hoje	5	0	3	0	1	42	3	3	1	0	1	31
não	17	14	0	0	9	8	26	0	1	2	4	9
passado	1	0	23	0	5	26	1	11	1	0	9	12
porque	4	2	7	0	2	35	6	2	9	1	0	22
sim	21	16	2	2	10	2	15	0	0	3	2	17
tchau	13	4	3	1	9	0	20	0	0	1	27	11
vontade	8	3	27	1	1	6	0	1	1	1	3	38

Considerando-se a segunda estratégia de divisão dos conjuntos de treinamento, validação e teste, após treinamento da RNC obteve-se um desempenho superior se comparada à primeira. A Tabela 4.8 apresenta os resultados obtidos pela segunda estratégia, comparando com a acurácia obtida pela primeira estratégia, discutida anteriormente. Algo a se destacar é que, de forma geral, a diferença entre acurácia para o conjunto de treinamento e teste foi menor se comparado aos resultados apresentados anteriormente. Um dos fatores que leva ao aumento da acurácia na segunda estratégia pode ser a repetição de cada sinal realizado pelos voluntários (cinco repetições); assim, o conjunto de treinamento apresentado à RNC, apesar da aleatoriedade na sua composição, pode possuir vídeo semelhante ao presente no conjunto de teste.

Tabela 4.8: Comparação das acurácias para reconhecimento dos sinais em vídeos usando as duas diferentes estratégias de divisão dos conjuntos

Grupos	Acurácia (Treinamento)	Acurácia (Teste)	Acurácia (1ª estratégia)
Grupo 1	0.6688	0.7536	0.43774
Grupo 2	0.6843	0.7860	0.45833
Grupo 3	0.7116	0.8076	0.42778
Grupo 4	0.5606	0.7263	0.31395
Todos	0.6304	0.7404	0.18985

Por fim, realizou-se uma breve análise de como as câmeras e os diferentes ângulos afetam a acurácia. Observou-se que não houve diferença significativa no desempenho da rede nesses diferentes contextos. Apesar das diferenças sutis na acurácia, não se percebeu um padrão que permita concluir que um ângulo de captura ou modelo de câmera é mais favorável que outro, pois, dependendo do grupo com que a RNC foi treinada, houve alternâncias de ângulo ou câmera nos melhores índices de acurácia. Contudo, devido aos poucos testes feitos nesse sentido, pouco se pode concluir. Uma investigação mais profunda deve ser realizada para averiguar como a qualidade da câmera e os ângulos de capturas afetam a tarefa de reconhecimento.

Capítulo 5
Conclusões

Este é um trabalho inicial visando à construção de um sistema tradutor de Libras-Português. Com os resultados obtidos acredita-se que tal sistema seja viável, considerando a evolução da velocidade de processamento em dispositivos como *smartphones*, que são plataformas interessantes para sistemas dessa natureza, além da velocidade de evolução dos métodos de classificação de imagens e vídeos, obtidas nos últimos anos. Porém para, isso, maiores pesquisas devem ser realizadas.

Apesar da acurácia ser de 19%, quando treinada para classificar todos os 48 sinais, seguindo a primeira estratégia de divisão dos conjuntos, os resultados mostraram que quando a RNC erra na classificação do sinal, muitas vezes classifica-o em um sinal que possui características semelhantes, como visto na matriz de confusão do grupo 1. Além disso, ao se considerar a acurácia nos índices *top-3* e *top-5*, percebeu-se um ganho significativo, chegando a uma acurácia superior a 50%, neste último índice. Com isso, a RNC mostra uma capacidade de generalização das características dos sinais presentes nos vídeos. Porém, em alguns casos, a rede classificou um sinal incorretamente, atribuindo a ela uma classe que muitas vezes possuía nenhuma ou poucas características do sinal esperado. Isso leva a crer que melhorias devem ser realizadas para construção de um sistema robusto de tradução.

O desempenho superior da RNC na classificação dos sinais da Libras, quando treinada com vídeos de todos os voluntários (segunda estratégia), mostrou que a rede é eficiente para traduzir sinais de intérpretes previamente conhecidos. Entretanto, na prática, tal estratégia não é válida, pois o sistema deverá analisar vídeos e reconhecer sinais realizados por indivíduos nunca visto antes. Assim, a primeira estratégia de treinamento da RNC é mais adequada para analisar o desempenho da rede.

Por fim, a RNC mostrou-se capaz de lidar com múltiplos ângulos e múltiplas câmeras, que afetam diretamente e de forma significativa as características dos dados de entrada. Isso é interessante, pois na prática o uso de um hipotético sistema de tradução, será feito através de

diferentes câmeras, além de diferentes contextos, como ângulos de captura diferentes. Porém, os vídeos presentes na base de dados possuem fundo estático, então mais cenários devem ser testados para avaliar o uso de uma RNC na classificação de sinais da Libras.

5.1 Trabalhos Futuros

Recomenda-se prosseguir este trabalho realizando mais experimentos com o intuito de:

1) testar outros métodos de classificação e extração de características;
2) empregar outras estratégias para aumentar artificialmente o conjunto de dados, além das transformações geométricas utilizadas neste trabalho;
3) melhorar a acurácia do reconhecimento dos sinais empregando outras arquiteturas de redes neurais;
4) utilizar e avaliar outras estratégias de separação da base de dados como a validação cruzada que, apesar de mais complexa, é mais robusta na avaliação da acurácia dos classificadores;
5) incorporar o contexto semântico de uma sentença para auxiliar no processo de classificação dos sinais;
6) aumentar o tamanho da base de dados incluindo novos sinais e voluntários, visando avaliar a escalabilidade do método proposto;
7) incorporar metadados aos vídeos registrando características físicas dos voluntários e dos sinais. Esses dados adicionais podem ser empregados pela RNC ou serem usados como elemento de agrupamento prévio dos vídeos na construção de conjuntos de treinamento, validação e teste.

Anexo A
Termo de Consentimento Livre Esclarecido

unioeste
Universidade Estadual do Oeste do Paraná

Pró-Reitoria de Pesquisa e Pós-Graduação
Comitê de Ética em Pesquisa – CEP

Aprovado na
CONEP em 04/08/2000

ANEXO I

TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO - TCLE

Título do Projeto: libras2texto: utilizando Redes Neurais Artificiais para reconhecer sinais da LIBRAS

Pesquisadores: Adair Santa Catarina (45)988137337, Renan Tashiro (45)998371311

Convidamos você a participar de nossa pesquisa que tem o objetivo de desenvolver um software de computador capaz de entender LIBRAS. Esperamos, com este estudo, ter um avanço na construção de um sistema robusto capaz de traduzir para a língua portuguesa os sinais da LIBRAS, assim diminuindo barreira da comunicação entre pessoas que possuem deficiência auditiva com a comunidade em geral. Para tanto, será necessário a coleta de vídeos com pessoas sinalizando em LIBRAS.

As pessoas que participarão da coleta de dados, caso apresentem algum problema de saúde física, serão socorridas por meio de atendimento especializado (SAMU). Caso aconteçam problemas de saúde de cunho psicológico, os participantes serão encaminhados para a Assistência Estudantil do Campus, que fará os devidos encaminhamentos para atendimento psicológico.

Sua identidade não será divulgada e seus dados serão utilizados apenas fins científicos. Você também não pagará nem receberá para participar do estudo. Além disso, você poderá cancelar sua participação na pesquisa a qualquer momento. No caso de dúvidas ou da necessidade de relatar algum acontecimento, você pode contatar os pesquisadores pelos telefones mencionados acima ou o Comitê de Ética pelo número 3220-3092.

Este documento será assinado em duas vias, sendo uma delas entregue ao sujeito da pesquisa.

( ) Declaro estar ciente do exposto e desejo participar da pesquisa.
( ) Autorizo a divulgação de minhas imagens, de forma anônima, em uma base de dados pública.

_____

(Assinatura)
(Nome do sujeito de pesquisa ou responsável)

Eu, Renan Tashiro, declaro que forneci todas as informações do projeto ao participante e/ou responsável.

_____

Renan Tashiro

Cascavel, _____ de _____ de _____.

Comitê de Ética em Pesquisa
Aprovado
05/04/18
Unioeste

Anexo B
Parecer do Comitê de Ética

UNIOESTE - CENTRO DE CIÊNCIAS BIOLÓGICAS E DA SAÚDE DA UNIVERSIDADE

PARECER CONSUBSTANCIADO DO CEP

DADOS DO PROJETO DE PESQUISA

Título da Pesquisa: libras2texto: utilizando Redes Neurais Artificiais para reconhecer sinais da LIBRAS

Pesquisador: Adair Santa Catarina
Área Temática:
Versão: 1
CAAE: 86108218.7.0000.0107
Instituição Proponente: Universidade Estadual do Oeste do Paraná/ UNIOESTE
Patrocinador Principal: Financiamento Próprio

DADOS DO PARECER

Número do Parecer: 2.588.637

Apresentação do Projeto:
Neste estudo será realizado uma pesquisa quantitativa para investigar a acurácia de uma rede neural artificial na tarefa de reconhecimento de sinais da LIBRAS. Para isso cerca de 20 a 100 voluntários serão amostrados. Estes voluntários serão gravados em vídeos curtos, realizando os sinais da LIBRAS, que serão utilizados para treinar a rede. Essa coleta será feito com estudantes e profissionais da Universidade do Oeste do Paraná (UNIOESTE) e espera-se com isso responder se técnicas modernas de aprendizado de máquina e visão computacional são boas o suficiente para tarefa de classificação de vídeos, usando como estudo de caso o reconhecimento de sinais da LIBRAS

Objetivo da Pesquisa:
Objetivo Primário:
Verificar se redes neurais artificiais são capazes de atingir um bom desempenho na tarefa de reconhecimento de sinais da LIBRAS com um quantidade razoável de dados

Avaliação dos Riscos e Benefícios:
Riscos:
A priori não possui, porém caso ocorra sinistros, imprevistos todo o aparato necessário de socorro ao voluntário será feito por meio de atendimento

Endereço: UNIVERSITARIA
Bairro: UNIVERSITARIO **CEP:** 85.819-110
UF: PR **Município:** CASCAVEL
Telefone: (45)3220-3272 **E-mail:** cep.prppg@unioeste.br

especializado (SAMU)

Benefícios:

Com esse estudo espera-se avançar na construção de um sistema robusto tradutor da LIBRAS para Língua Portuguesa.

Comentários e Considerações sobre a Pesquisa:

Indica ser importante para a área e para os envolvidos

Considerações sobre os Termos de apresentação obrigatória:

Presentes e adequados

Recomendações:

Sem recomendações

Conclusões ou Pendências e Lista de Inadequações:

Sem pendências

Este parecer foi elaborado baseado nos documentos abaixo relacionados:

Tipo Documento	Arquivo	Postagem	Autor	Situação
Informações Básicas do Projeto	PB_INFORMAÇÕES_BÁSICAS_DO_PROJETO_1094439.pdf	22/03/2018 16:19:30		Aceito
Declaração de Instituição e Infraestrutura	responsavel_campo.pdf	22/03/2018 16:18:22	Adair Santa Catarina	Aceito
Declaração de Pesquisadores	inicio_coleta_dados.pdf	22/03/2018 16:18:01	Adair Santa Catarina	Aceito
Declaração de Pesquisadores	compromisso_dados.pdf	22/03/2018 16:17:51	Adair Santa Catarina	Aceito
Projeto Detalhado / Brochura Investigador	ModeloProjetoTCC.pdf	22/03/2018 16:15:50	Adair Santa Catarina	Aceito
TCLE / Termos de Assentimento / Justificativa de Ausência	Modelo_de_TcleNOVO.pdf	22/03/2018 16:14:40	Adair Santa Catarina	Aceito
Folha de Rosto	folha_rosto.pdf	22/03/2018 16:12:57	Adair Santa Catarina	Aceito

Situação do Parecer:

Aprovado

Necessita Apreciação da CONEP:
Não

CASCAVEL, 09 de Abril de 2018

Assinado por:
Dartel Ferrari de Lima
(Coordenador)

Endereço: UNIVERSITARIA
Bairro: UNIVERSITARIO **CEP:** 85.819-110
UF: PR **Município:** CASCAVEL
Telefone: (45)3220-3272 **E-mail:** cep.prppg@unioeste.br

Apêndice A

Trechos de Código da Implementação

O Quadro 1 mostra um trecho do código que define um bloco S3D, escrito em Python e utilizando o Keras. O parâmetro *nb_filters* define a quantidade de filtros que o bloco utiliza; *padding* com valor *same* indica que após realizar a operação de convolução sobre os dados de entrada da camada, a sua dimensão original é preservada. A função de não-linear utilizada foi a *elu* (*exponential linear unit*). Como entrada do bloco se tem os dados processados pela camada anterior. Como retorno, se tem o vetor de características, após a entrada ser processado por cada filtro concatenados. Cada xi representa um filtro, note que no caso do $x2$, por exemplo, que o filtro está definido através de várias convoluções separados. XIE (2017) explica em detalhes a razão, porém em resumo essa separação auxilia em deixar a arquitetura menor em número de parâmetros, afetando pouco o desempenho,

Quadro 1 - Definição de um bloco S3D

```
01 | x1 = Conv3D(nb_filters, (1, 1, 1))(tensor)
02 |
03 | x2 = Conv3D(nb_filters, (1, 1, 1))(tensor)
04 | x2 = Conv3D(nb_filters, (1, 2, 2), padding='same')(x2)
05 | x2 = Conv3D(nb_filters, (2, 1, 1), padding='same')(x2)
06 | x2 = Activation('elu')(x2)
07 |
08 | x3 = Conv3D(nb_filters, (1, 1, 1))(tensor)
09 | x3 = Conv3D(nb_filters, (1, 3, 3), padding='same')(x3)
10 | x3 = Conv3D(nb_filters, (3, 1, 1), padding='same')(x3)
11 | x3 = Activation('elu')(x3)
12 |
13 | x5 = Conv3D(nb_filters, (1, 1, 1))(tensor)
14 | x5 = Conv3D(nb_filters, (1, 5, 5))(x5)
15 | x5 = Conv3D(nb_filters, (5, 1, 1))(x5)
16 | x5 = Activation('elu')(x5)
17 |
18 | return concatenate([x1, x2, x3, x5])
```

O Quadro 2 apresenta o código que define a arquitetura utilizada para extrair as características espaciais e temporais dos dados de entrada. O primeiro parâmetro da função S3D representa o número de filtros utilizados e o segundo argumento representa os dados de saída da camada anterior. O *spacial_features* representa os dados de saída da MobileNetV2. O *Flatten* transforma os dados matriciais em um vetor unidimensional, para que possa ser utilizado pelo classificador.

Quadro 2 - Extrator de características espaço-temporais

```
01 | x = S3D(32, spacial_features)
02 | x = S3D(32, x)
03 | x = S3D(32, x)
04 | x = MaxPooling3D()(x)
05 |
06 | x = S3D(64, x)
07 | x = S3D(64, x)
08 | x = S3D(64, x)
09 |
10 | x = S3D(128, x)
11 | x = S3D(128, x)
12 | x = S3D(128, x)
13 | x = MaxPooling3D()(x)
14 |
15 | return Flatten()(X)
```

O Quadro 3 mostra um trecho do código que define o classificador, que utiliza o vetor de características fornecidos pelas camadas anteriores para retornar, como saída, um vetor que contém a probabilidade de um vídeo possuir um determinado sinal. Como retorno, esse trecho de código tem um objeto que representa a arquitetura configurada.

Quadro 3 - Classificador da arquitetura utilizada

```
01 | classifier = Dense(256)(x)
02 | classifier = Activation('elu')(classifier)
03 |
04 | classifier = Dense(256)(classifier)
05 | classifier = Activation('elu')(classifier)
06 |
07 | classifier = Dense(nb_classes)(classifier)
08 | classifier = Activation('softmax')(classifier)
09 |
10 | return Model(inputs=[video_input], outputs=[classifier])
```

Apêndice B

Acurácia ao Longo do Treinamento

As figuras a seguir mostram como a acurácia mudou ao longo do treinamento das RNA. O eixo $x$representa o número de épocas e o eixo ya acurácia. Os pontos representam a acurácia obtida com o conjunto de treinamento e a linha contínua a acurácia obtida com o conjunto de validação.

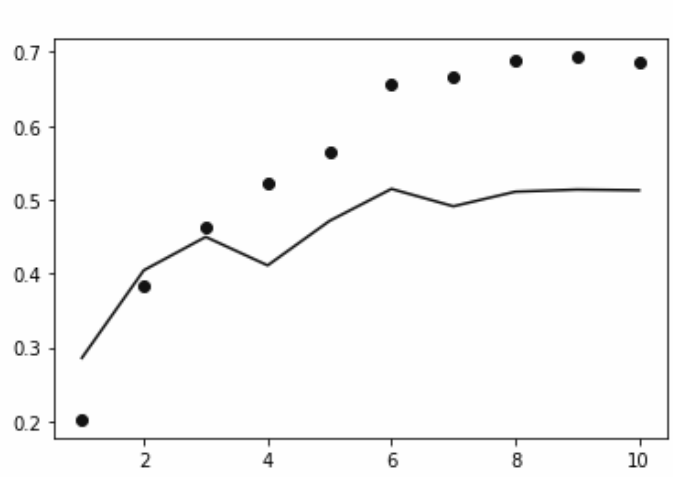

Figura 1: Acurácia do conjunto de treinamento e validação (Grupo 1)

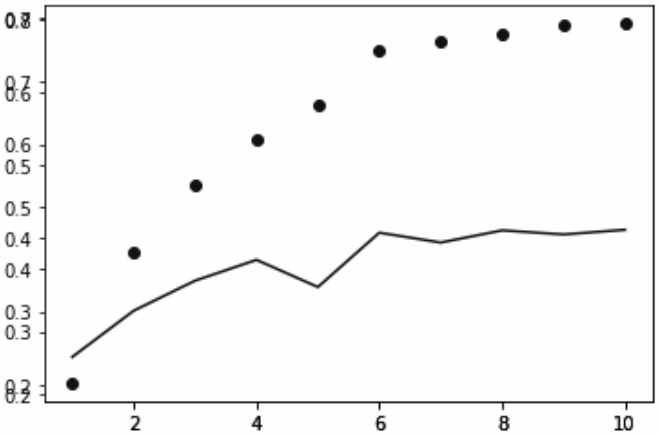

Figura 2: Acurácia do conjunto de treinamento e validação (Grupo 2)

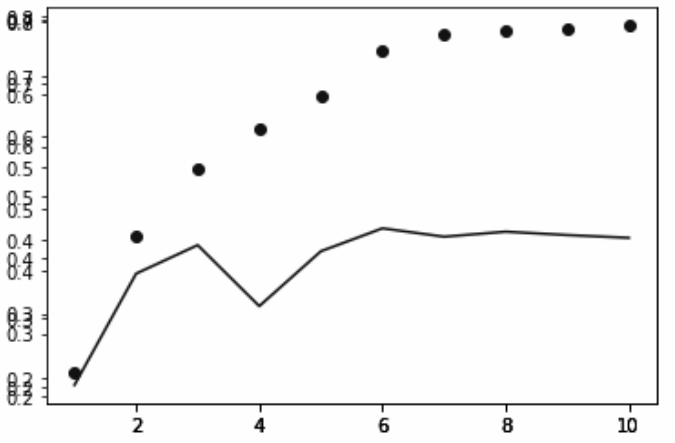

Figura 3: Acurácia do conjunto de treinamento e validação (Grupo 3)

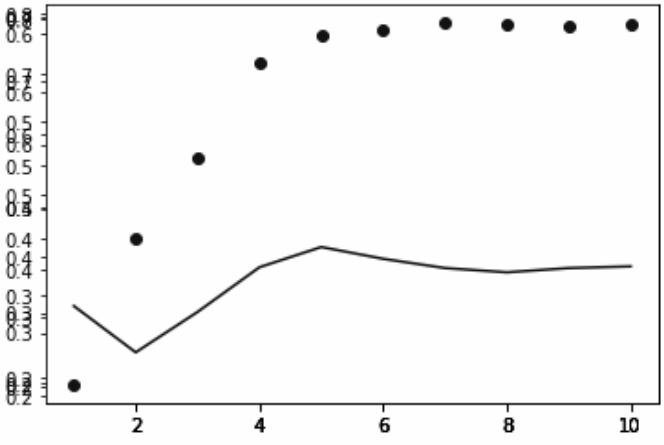

Figura 4: Acurácia do conjunto de treinamento e validação (Grupo 4)

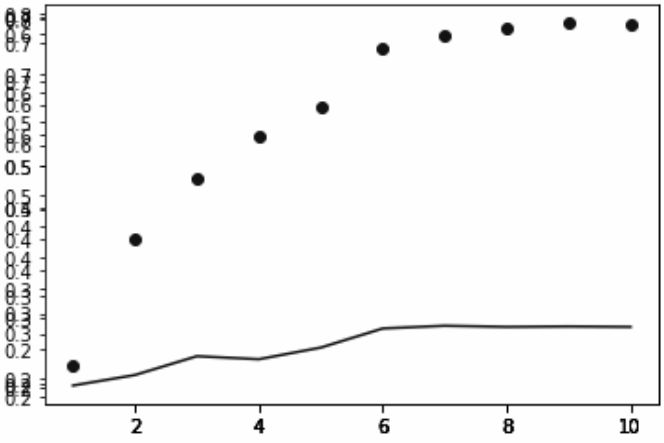

Figura 5: Acurácia do conjunto de treinamento e validação (Todos)

Apêndice C

Matriz de Confusão para Todos os Sinais

Os quadros a seguir mostram a matriz de confusão de uma RNA treinada com todos os sinais presentes na base de dados.

Quadro 1: Matriz de confusão (Parte 1)

Sinais	agosto	ainda	assistir	avisar	avisar-me	bom	branco	casa	como	dia	dúvida	educado
agosto	41	0	0	0	1	1	1	0	0	0	0	1
ainda	0	16	17	3	0	2	0	0	9	0	0	0
assistir	0	15	45	6	0	0	0	1	5	5	4	0
avisar	2	0	6	16	3	22	1	0	0	0	0	0
avisar-me	4	4	0	3	21	3	0	0	0	0	0	0
bom	1	2	0	7	1	13	3	0	2	0	0	2
branco	0	0	0	1	2	0	56	0	0	0	0	9
casa	0	0	2	3	3	0	1	9	0	0	5	1
como	0	1	14	0	0	0	0	0	9	0	0	0
dia	0	2	22	6	4	3	0	0	6	2	0	1
dúvida	0	0	2	0	0	0	0	8	0	0	32	0
educado	0	0	0	0	0	0	33	0	0	0	0	41
ele-ela	0	0	3	2	1	4	0	0	0	0	0	0
entender	0	0	6	0	0	0	0	0	0	18	0	0
entender-não	0	0	6	0	1	0	1	0	0	20	0	0
esquecer	0	0	6	0	0	0	0	2	0	14	0	0
estuda	0	0	0	0	0	0	13	0	1	0	0	1
eu	0	5	6	0	3	1	4	0	6	6	1	0
faculdade	0	6	29	1	0	1	0	0	0	5	0	0
família	0	0	0	0	1	2	0	0	0	0	0	0
futuro	0	4	4	3	1	11	0	0	2	1	0	0
gostar	7	0	5	1	4	0	5	0	2	0	4	1
hoje	1	4	0	3	2	0	0	0	4	0	1	0
idade	2	0	21	0	1	0	0	0	0	0	0	2

Quadro 2: Matriz de confusão (Parte 2)

Sinais	ele-ela	entender	entender-não	esquecer	estuda	eu	faculdade	família	futuro	gostar	hoje	idade
agosto	0	0	0	0	3	0	0	0	0	21	0	0
ainda	1	0	0	0	0	7	0	0	0	1	0	0
assistir	0	0	0	0	0	0	0	0	1	0	0	0
avisar	2	0	1	0	0	0	1	3	2	6	0	0
avisar-me	1	0	0	0	2	1	0	1	0	8	0	0
bom	2	0	0	0	3	4	0	4	18	1	0	0
branco	0	1	0	0	6	0	0	0	0	0	0	0
casa	0	0	1	0	8	0	0	9	0	1	0	0
como	0	0	0	0	0	0	0	0	0	0	0	0
dia	3	0	0	1	1	5	3	1	1	0	0	0
dúvida	0	0	0	0	0	0	0	3	0	0	0	0
educado	0	0	0	0	1	0	0	0	0	0	0	0
ele-ela	17	0	0	0	0	1	1	1	26	0	0	0
entender	1	26	21	0	0	0	0	0	0	0	0	16
entender-não	1	5	36	4	1	0	0	0	0	0	0	5
esquecer	0	0	0	48	0	0	1	0	0	0	0	0
estuda	0	0	0	0	19	1	0	9	0	0	0	0
eu	0	0	0	0	0	8	0	0	2	0	0	0
faculdade	0	0	0	1	0	2	29	0	1	0	0	0
família	2	0	0	0	1	1	0	60	4	1	0	0
futuro	9	0	0	0	0	3	0	3	15	1	0	0
gostar	0	0	0	0	0	0	0	1	1	17	0	0
hoje	2	0	0	0	7	1	0	2	0	3	3	0
idade	0	1	6	0	4	3	1	0	0	1	0	5

Quadro 3: Matriz de confusão (Parte 3)

Sinais	ler	local	meu	não	noite	nome-meu	nome-seu	nos	oi	passado	passear	pessoa
agosto	2	2	9	0	0	1	0	1	0	0	2	0
ainda	0	0	3	1	0	0	0	5	0	0	0	0
assistir	1	0	0	0	0	0	0	0	0	0	0	0
avisar	0	0	1	0	0	2	0	2	0	0	0	0
avisar-me	0	1	10	0	0	1	0	9	0	0	0	0
bom	2	0	4	3	1	1	0	1	0	0	0	0
branco	1	2	0	0	7	0	1	0	0	0	0	0
casa	17	0	2	0	0	1	0	0	0	0	0	0
como	0	0	0	1	0	0	0	0	0	0	0	0
dia	2	0	3	0	0	0	0	5	1	0	0	0
dúvida	11	0	0	0	0	0	0	0	0	1	0	0
educado	1	0	1	0	6	1	0	0	0	1	0	0
ele-ela	0	0	2	1	0	3	0	4	0	0	0	0
entender	0	0	0	0	0	0	0	0	1	0	0	1
entender-não	0	0	0	0	1	0	0	0	1	0	1	7
esquecer	0	0	0	0	0	0	0	0	0	0	0	19
estuda	14	1	0	0	4	0	0	0	0	0	0	0
eu	3	0	3	0	0	1	0	0	0	0	0	0
faculdade	0	0	0	0	0	1	0	0	0	0	1	0
família	0	1	0	0	0	0	0	1	0	0	0	0
futuro	0	0	0	2	0	0	0	6	0	1	0	0
gostar	4	0	7	0	0	0	0	2	0	1	0	0
hoje	13	0	3	1	0	0	0	0	0	2	1	0
idade	5	0	2	2	4	0	0	0	3	0	5	0

Quadro 4: Matriz de confusão (Parte 4)

Sinais	porque	quente	rápido	sentimento	sim	tarde	tchau	ter	trabalha	tv	você	vontade
agosto	0	0	2	0	0	2	0	0	0	0	0	0
ainda	1	0	0	0	2	0	4	9	0	0	8	1
assistir	0	0	0	0	5	0	0	0	0	0	1	1
avisar	0	1	3	0	2	1	0	1	0	0	12	0
avisar-me	0	6	0	0	0	2	0	1	0	0	3	0
bom	2	0	1	1	3	1	1	0	0	0	6	0
branco	0	1	0	0	2	0	0	3	0	0	0	0
casa	7	0	0	0	0	0	0	0	0	4	0	16
como	1	0	0	0	1	0	0	18	0	0	0	0
dia	0	0	2	0	2	4	0	4	0	0	4	1
dúvida	3	0	0	0	0	0	0	1	0	13	0	16
educado	0	0	0	1	0	0	0	1	3	0	0	0
ele-ela	0	2	0	0	1	1	2	10	0	0	8	0
entender	0	0	0	0	0	0	0	0	0	0	0	0
entender-não	0	0	0	0	0	0	0	0	0	0	0	0
esquecer	0	0	0	0	0	0	0	0	0	0	0	0
estuda	19	0	0	0	1	0	0	4	0	0	1	2
eu	3	0	0	10	1	1	4	14	0	0	5	3
faculdade	0	0	0	0	1	0	0	1	0	0	10	1
família	0	0	0	0	0	6	0	1	0	1	3	5
futuro	0	0	0	0	0	1	2	1	0	0	18	2
gostar	2	0	0	2	0	0	5	9	0	5	0	5
hoje	8	0	1	0	1	0	1	4	1	5	4	12
idade	2	0	2	2	0	5	2	6	0	1	2	0

Quadro 5: Matriz de confusão (Parte 5)

Sinais	agosto	ainda	assistir	avisar	avisar-me	bom	branco	casa	como	dia	dúvida	educado
ler	0	0	4	0	0	0	5	0	2	0	2	1
local	3	0	0	0	0	0	10	0	0	0	0	1
meu	1	1	8	0	4	0	6	0	4	0	1	5
não	0	10	2	2	4	1	0	0	14	0	0	0
noite	1	0	0	0	0	1	3	0	0	0	3	3
nome-meu	5	0	0	1	5	0	9	0	0	0	0	0
nome-seu	3	0	0	0	11	5	6	0	0	0	0	0
nos	0	2	6	0	2	0	2	0	0	0	0	0
oi	8	0	0	5	7	2	3	0	0	0	0	0
passado	0	0	0	0	0	1	1	5	0	0	8	0
passear	0	0	5	0	0	0	0	5	1	0	4	0
pessoa	0	0	4	0	0	0	0	0	0	7	0	0
porque	0	7	0	2	3	3	0	1	2	0	2	0
quente	2	0	0	2	1	1	4	3	0	2	0	0
rápido	8	0	0	5	11	1	4	0	0	1	0	0
sentimento	23	0	0	4	7	1	0	1	0	0	0	3
sim	0	7	14	3	0	3	0	0	17	0	0	0
tarde	0	1	3	1	1	1	9	0	2	0	0	2
tchau	0	9	10	7	3	3	0	0	8	0	0	0
ter	5	7	5	0	2	2	2	0	6	0	1	1
trabalha	0	0	0	1	0	1	3	1	0	0	1	2
tv	0	1	0	0	0	0	0	2	0	0	7	0
você	0	1	8	0	0	2	1	0	2	0	0	0
vontade	0	9	6	3	1	0	0	4	0	0	12	0

Quadro 6: Matriz de confusão (Parte 6)

Sinais	ele-ela	entender	entender-não	esquecer	estuda	eu	faculdade	família	futuro	gostar	hoje	idade
ler	0	0	0	0	2	0	0	2	0	0	6	0
local	1	0	0	0	5	0	0	12	0	5	3	0
meu	2	0	0	0	0	1	0	2	1	2	0	0
não	4	0	0	0	2	2	1	1	2	2	0	0
noite	0	0	0	0	8	2	0	8	0	2	1	0
nome-meu	0	0	0	0	2	6	0	4	0	3	0	0
nome-seu	0	0	0	0	1	6	0	4	0	4	0	0
nos	2	0	0	0	0	0	4	2	13	2	0	0
oi	2	0	0	0	3	3	0	4	1	10	1	0
passado	0	0	0	0	3	1	0	14	4	1	5	0
passear	0	0	0	0	0	0	0	2	0	0	11	1
pessoa	0	0	1	31	0	0	2	0	0	0	0	2
porque	4	0	0	0	12	7	0	3	0	2	2	0
quente	1	0	0	0	1	0	0	0	0	0	0	0
rápido	0	0	0	0	3	0	0	3	1	5	0	0
sentimento	0	0	0	0	0	4	0	0	0	5	0	1
sim	5	0	0	0	0	3	1	1	0	0	0	2
tarde	3	0	0	0	2	3	0	13	6	0	2	0
tchau	1	0	0	0	0	4	0	0	0	4	0	2
ter	0	0	0	0	0	2	0	0	0	0	0	1
trabalha	0	0	0	0	6	1	0	3	0	0	0	0
tv	0	0	0	0	0	0	0	23	0	1	10	0
você	4	0	0	0	1	1	0	0	15	0	1	0
vontade	1	0	0	0	0	1	0	7	0	3	0	0

Quadro 7: Matriz de confusão (Parte 7)

Sinais	ler	local	meu	não	noite	nome-meu	nome-seu	nos	oi	passado	passear	pessoa
ler	30	0	2	0	0	0	0	0	0	1	1	0
local	14	6	0	0	1	1	0	0	1	4	0	0
meu	10	0	5	0	0	0	0	0	0	0	0	0
não	2	0	2	2	0	0	0	5	0	0	1	0
noite	16	0	1	0	14	0	0	0	0	1	0	0
nome-meu	4	0	0	7	0	25	9	0	0	0	0	0
nome-seu	3	0	2	2	1	12	12	1	2	0	0	0
nos	4	0	0	0	0	2	0	27	0	0	0	0
oi	4	0	2	3	1	8	1	3	2	0	0	0
passado	5	1	0	1	2	0	0	0	0	15	0	0
passear	6	0	0	0	0	0	0	0	0	0	32	0
pessoa	0	0	0	0	0	0	0	0	10	0	0	33
porque	4	0	7	1	2	0	0	2	0	2	0	0
quente	0	0	1	0	2	2	6	13	0	0	0	0
rápido	2	4	1	0	0	2	5	2	0	0	0	0
sentimento	0	2	15	0	0	0	0	0	0	0	0	0
sim	1	0	4	1	0	1	0	4	0	0	2	0
tarde	12	0	0	2	1	1	0	5	0	0	0	0
tchau	1	0	4	1	0	0	0	5	0	0	0	0
ter	0	0	6	8	0	0	0	0	0	0	0	0
trabalha	37	2	0	0	2	0	1	0	0	2	2	0
tv	8	2	0	0	0	0	0	0	0	1	2	0
você	3	0	0	1	2	2	0	8	1	0	0	0
vontade	2	0	2	0	0	0	0	0	0	5	0	0

Quadro 8: Matriz de confusão (Parte 8)

Sinais	porque	quente	rápido	sentimento	sim	tarde	tchau	ter	trabalha	tv	você	vontade
ler	13	0	0	1	5	2	0	4	0	3	0	3
local	9	0	5	0	0	0	0	0	0	9	0	0
meu	2	2	0	7	0	0	3	10	0	4	2	7
não	0	0	4	1	1	1	1	17	0	0	3	3
noite	19	0	0	0	0	0	0	6	0	1	0	0
nome-meu	2	0	5	0	0	0	1	1	0	0	1	0
nome-seu	3	0	3	2	0	0	2	2	0	1	2	0
nos	0	6	2	0	2	0	4	3	0	1	1	3
oi	1	1	3	0	0	1	2	0	1	0	7	1
passado	3	0	0	0	0	6	1	1	0	11	0	1
passear	1	1	0	0	4	2	3	1	1	8	0	2
pessoa	0	0	0	0	0	0	0	0	0	0	0	0
porque	9	0	1	1	0	0	0	4	0	0	2	5
quente	0	34	4	0	0	2	0	0	0	0	4	0
rápido	0	3	23	0	0	0	0	0	0	1	0	0
sentimento	0	1	1	17	0	0	0	3	2	0	0	0
sim	0	0	0	0	2	1	1	11	0	0	6	0
tarde	4	0	0	0	4	10	1	0	0	0	1	0
tchau	0	0	0	0	1	0	10	8	0	0	7	2
ter	1	0	0	3	1	1	2	33	0	0	0	1
trabalha	11	0	4	0	0	0	0	0	4	6	0	0
tv	0	0	0	0	0	2	1	1	0	28	0	1
você	0	1	0	0	0	2	1	9	0	0	20	4
vontade	1	0	0	0	1	0	0	1	0	8	2	21

Referências Bibliográficas

ABADI, M. *Tensorflow: Large-scale machine learning on heterogeneous systems*, 2015. Disponível em: <https://tensorflow.org>. Acesso em: jun. 2018.

ACESSIBILIDADE BRASIL. *Dicionário da Língua Brasileira de Sinais V3 - 2011*. Disponível em: <https://www.acessibilidadebrasil.org.br/libras_3>. Acesso em: jun. 2018.

BRASIL. Lei nº 10.436, de 24 de abril de 2002. *Dispõe sobre a Língua Brasileira de Sinais - Libras e dá outras providências*. Diário Oficial [da] República Federativa do Brasil, Brasília, DF, 25 de abr. 2002. Disponível em: <http://www.planalto.gov.br/ccivil_03/leis/2002/l10436.htm>. Acesso em: jun. 2018.

BRITO, L. F.; QUADROS, R. M.; FELIPE, T. A. *Língua Brasileira de Sinais - LIBRAS*, 2017. Disponível em: <https://www.librasgerais.com.br/materiais-inclusivos/downloads/gramatica-libras.pdf>. Acesso em: jun. 2018.

CHOLLET, F. *et al*. *Keras*, 2015. Disponível em: <https://keras.io>. Acesso em: jun. 2018.

DESHPANDE, A. *A Beginner's Guide To Understanding Convolutional Neural Networks*, 2016. Disponível em: <https://adeshpande3.github.io/adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/>. Acesso em: out. 2018.

DONAHUE, J. *et al*. Long-term Recurrent Convolutional Networks for Visual Recognition and Description. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 2017. p. 677-691.

GAO, W.; FANG, G.; CHEN, Y. A Chinese sign language recognition system based on SOFM/SRN/HMM. *Pattern Recognition*, 2014. p. 2389-2404.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE., A. *Deep Learning*. MIT Press, 2016.

IBGE. CENSO DEMOGRÁFICO 2010: *Características gerais da população, religião e pessoas com deficiência*. Rio de Janeiro: IBGE, 2012. Acompanha 1 CD-ROM. Disponível em: <https://biblioteca.ibge.gov.br/visualizacao/periodicos/94/cd_2010_religiao_deficiencia.pdf>. Acesso em: jun. 2018.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. ImageNet Classification with Deep Convolutional Neural Network. NIPS 2012, *Advances in Neural Information Processing Systems* 25. Curran Associates, Inc., 2012. p. 1097-1105.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep Learning. *Nature* 512, mai. 2015. p. 436-444.

LOKHANDE, P. Data Gloves for Sign Language Recognition System. *International Journal of Computer Applications*, 2015. p 11-14.

MONTEIRO, C. H. A. *et al*. Um sistema de baixo custo para reconhecimento de gestos em LIBRAS utilizando visão computacional. In: SBrT, 34, 2016,Santarém. *Anais do XXXIV SBrT*, 2016. p 349-352.

NEIVA, D. H; ZANCHETTIN, C. Gesture recognition: A review focusing on sign language in a mobile context. *Expert Systems with Applications*, vol. 103, 2018. p. 159-183.

NIELSEN, M. A. *Neural Networks and Deep Learning*. Determination Press, 2015.

OpenCV Team. *OpenCV: Open Source Computer Vision Library*. Disponível em: <https://github.com/opencv/opencv>. Acesso em: jun. 2018.

Python Core Team. *Python: A dynamic, open source programming language*. Python Software Foundation, 2018. Disponível em <https://www.python.org>. Acesso em: jul. 2018.

SANDLER, M. *et al*. MobileNetV2: Inverted Residuals and Linear Bottlenecks. *arXiv preprint*, 2018

STROBEL, K. L; FERNANDES, S. *Aspectos Linguísticos da LIBRAS: Língua Brasileira de Sinais*. Curitiba: SEED/SUED/DEE, 1998.

SUHARJITO *et al.* Sign Language Recognition Application Sustems for Deaf-Mute People: A Review Based on Input-Process-Output. *Procedia Computer Science*, vol 116, 2017. p 441-448.

TEODORO, B. T. *Sistema de Reconhecimento Automático de Língua Brasileira de Sinais*. Dissertação (Mestrado) - Escola de Artes, Ciências e Humanidades da Universidade de São Paulo. São Paulo, 2015.

WANG, H. *et al*. Fast sign language recognition benefited from low rank approximation. *Conference and Workshops on Automatic Face and Gesture Recognition*, ed. 11, 2015.

ZHANG, L. G. *et al*. A Vision-Based Sign Language Recognition System. *Conference on Multimodal Interfaces*, ed. 6, 2004. p. 198-204.

XIE, S. *et al*. Rethinking Spatiotemporal Feature Learning For Video Understanding. *CoRR*, 2017.