UNIVERSIDADE ESTADUAL DO MARANHÃO - UEMA
CENTRO DE CIÊNCIAS TECNOLÓGICAS - CCT
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DA COMPUTAÇÃO E SISTEMA
MESTRADO EM ENGENHARIA DA COMPUTAÇÃO E SISTEMAS

ALEX LUIS DA COSTA ALEXANDRE

PLATAFORMA PARA RECONHECIMENTO DA LINGUAGEM BRASILEIRA DE SINAIS UTILIZANDO KINECT

São Luís

2019

ALEX LUIS DA COSTA ALEXANDRE

PLATAFORMA PARA RECONHECIMENTO DA LINGUAGEM BRASILEIRA DE SINAIS UTILIZANDO KINECT

Projeto de pesquisa ao programa do Mestrado Profissional de Engenharia da Computação e Sistemas da Universidade Estadual do Maranhão como parte dos requisitos para obtenção do título de Mestre em Engenharia da Computação.

Orientador: Prof. Dr. Mauro Sergio Silva Pinto
Coorientador: Prof. Msc. Denner Robert Rodrigues Guilhon

São Luís

2019

Alexandre, Alex Luís da Costa.
Plataforma para reconhecimento da linguagem brasileira de sinais utilizando Kinect / Alex Luis da Costa Alexandre. – São Luís, 2019.
63 f.

Dissertação (Mestrado) – Curso de Engenharia de Computação e Sistemas, Universidade Estadual do Maranhão, 2019.

Orientador: Prof. Dr. Mauro Sérgio Silva Pinto.

1.LIBRAS. 2.Kinect. 3.Processamento de imagens. 4.Segmentação. I.Título.

CDU: 004.4’4

ALEX LUIS DA COSTA ALEXANDRE

PLATAFORMA PARA RECONHECIMENTO DA LINGUAGEM BRASILEIRA DE SINAIS UTILIZANDO KINECT

Projeto de pesquisa ao programa do Mestrado Profissional de Engenharia da Computação e Sistemas da Universidade Estadual do Maranhão como parte dos requisitos para obtenção do título de Mestre em Engenharia da Computação.

Trabalho aprovado. São Luís, 30 de Setembro de 2019 :

BANCA EXAMINADORA

Prof. Dr. Mauro Sergio Silva Pinto
(Orientador)
Universidade Estadual do Maranhão

Prof. Msc. Denner Robert Rodrigues Guilhon (Co-Orientador)
Universidade Estadual do Maranhão

Prof. Dr. André Borges Cavalcante
Universidade Federal do Maranhão

Prof. Dr. Reinaldo de Jesus da Silva
Universidade Estadual do Maranhão

São Luís
2019

"*Porque!! um dia é preciso parar de sonhar,*
tirar os planos das gavetas e,
de algum modo, começar...".
(Amyr Klink)

Resumo

O trabalho apresenta uma forma alternativa na comunicação entre pessoas que só conseguem se comunicar pela linguagem brasileiras de sinais (LIBRAS) e pessoas que não tem domínio da língua. A maneira escolhida para facilitar esta comunicação é com a utilização do sensor Kinect (usado em vídeo game Xbox 360 e One) que dispõem de tecnologias de captura de imagens em RGB e imagens em profundidade (profundidade). Estas facilitam o processamento em software como Matlab.

No trabalho são apresentadas técnicas de aquisição, segregação e extração de informações da imagens além de sua aplicação em conjunto de atores com idades, sexo, tonalidade da pele e estaturas diferentes.

Em diversos trabalhos são apresentadas técnicas de segmentação usando as imagens de profundidade. No sensor Kinect 360 isso é relativamente simples de realizar, já o sensor Kinect One apresenta dificuldades na aplicação, devido a diferença nas resoluções. Nosso trabalho trás uma abordagem que resolve de forma satisfatória este problema.

Este trabalho também apresenta uma analise bibliográfica das publicações mais utilizadas. Ele auxilia pesquisadores iniciantes com as técnicas e os comandos utilizados no Matlab.

Palavras chaves:: LIBRAS, kinect, processamento de imagens, segmentação.

Abstract

This work presents an alternative form of communication between people who can only communicate through the Brazilian language of signs and people who do not have command of the language. The way presented is with the use of the sensor Kinect (used in video game Xbox one) which have image capture technologies that facilitate processing in software such as Matlab.

This work presents techniques for the acquisition, segregation and extraction of information from the images, as well as their application to actors of different ages, gender, skin tone and height.

In several works are presented segmentation techniques using profundidade images. In Kinect 360 sensor this is relatively simple to accomplish, while Kinect One sensor has difficulties in application due to the difference in resolutions. Our work takes an approach that satisfactorily solves this problem.

This work also presents a bibliographical analysis of the most used publications. It assists novice researchers with the techniques and commands used in Matlab.

Keywords: LIBRAS, kinect, image processing, segmentation.

Lista de ilustrações

Lista de tabelas

Sumário

1 INTRODUÇÃO

Desde os tempos primitivos as pessoas tinham dificuldade na comunicação entre eles. Devido a necessidade de criar utensílios de caça e proteção o homem através de gestos e repetição do processo criou a forma mais primitiva de linguagem.

Na comunicação gestual brasileira temos um grupo de pessoas com deficiência auditiva que possuem uma língua própria, chamada Língua Brasileira de Sinais ou simplesmente LIBRAS. A língua de sinais não é igual ao português, tem morfologia e sintaxe próprias, é um idioma brasileiro independente. Como qualquer outra língua, é preciso estudar a gramática e estruturação da frase pra dominar esta língua. A LIBRAS é reconhecida por lei como forma de comunicação e expressão das comunidades surdas do Brasil.

Os avanços da tecnologia com relação a captura de imagens têm viabilizado técnicas de visão computacional e reconhecimento de padrões em diferentes áreas da atividade humana (MENDONÇA., 2013). Infelizmente gestos e expressões corporais emitidas pelo homem ainda não são igualmente assimiláveis pelos atuais sistemas de computação. Nesse sentido faz-se necessário trabalhos de pesquisas de hardware e software para áreas como a interação humano-computador ou de aplicação de interfaces naturais com o usuário (JUNIOR., 2014).

1.1 Objetivo

1.1.1 Objetivo geral

Criar uma plataforma para reconhecimentos gestos e expressões corporais estáticas da linguagem brasileira de sinais a partir da captura de imagens do sensor de Microsoft Kinect.

1.1.2 Objetivos específicos

- Adquirir uma base de dados com imagens na língua brasileira de sinais com pessoas de idade, sexo e estaturas diversas;
- Criar meios para melhorar a comunicação entre pessoas com deficiência auditiva e pessoas sem deficiência;
- Apresentar as diversas técnicas no processamento de imagens com resultados para uma melhor escolha de qual técnica a ser escolhida.

1.2 Justificativa

A comunicação entre as pessoas é de fundamental importância para a construção de uma sociedade. As pessoas diariamente tem necessidades de interagir entre elas. Seja num atendimento médico, ou na compra de um produto na loja, ou num embarque no aeroporto, etc..

No meio dessa sociedade existem um grupo de pessoas que tem dificuldades de comunicação com as demais. Esse grupo de pessoas com deficiência auditiva empregam um esforço enorme para entender o que as outras pessoas falam, mesmo sem poder ouvir elas se aprimoraram em fazer leituras labiais. Mas no momento que elas passam da figura de receptor e se tornam os emissores na comunicação, estas encontram dificuldades. A linguagem que parte desse grupo tem habilidade em usar, a LIBRAS, não é de domínio da maioria da população brasileira.

Com a crescente pesquisa da comunidade acadêmica no processamento de imagens para o **reconhecimento de gestos**, esta trouxe avanços na interação humano-computador (IHC), o que tende a tornar viável a interação com o computador por meios de gestos (ANJO, 2013).

1.3 Estrutura do trabalho

Um sistema artificial de visão é um sistema computadorizado capaz de adquirir, processar e interpretar imagens em tempo real (FILHO; NETO., 1999) . A figura 1 mostra um diagrama de blocos de uma SVA.

Figura 1 – Um Sistema de Visão Artificial (SVA) e suas principais etapas

Fonte: (FILHO; NETO., 1999)

1.3.1 Aquisição

Uma das etapas mais importantes é a captura de imagem, nesta etapa consiste a entrada de dados que serão tratados e analisados para o processamento da imagem

O rastreio das mãos não é uma tarefa fácil, as mãos são objetos que se deformam mudando de pose e posição no espaço. O movimento das mãos podem ocultar o movimento da outra e da face. Imagens de profundidade são utilizadas para diminuir este problema (CORREIA, 2013).

A proposta é criar uma interação humano-computador (IHC) utilizando recursos de visão computacional, sem utilização de dispositivos como teclado, mouse.

Para aquisição de imagens será utilizado o sensor Microsoft Kinect utilizado no videogame Xbox. O kinect tem dois canais de video: imagens RGB e imagens de profundidade. Para realizar o processamento de imagem de profundidade, é utilizado o sensor infravermelho projetando uma imagem feita por padrões de difração pseudoaleatórios estáticos (um holograma gerado computacionalmente) sobre a cena. (JUNIOR., 2014).

1.3.2 Pré-processamento

Um imagem pode apresentar diversas imperfeições, tais como: pixel ruidosos, contraste e/ou brilho inadequado, caracteres interrompidos. A função desta etapa é aprimorar a imagem para as fases subsequentes. As operações efetuadas são de baixos níveis porque trabalham com a intensidade dos pixels. A imagem resultante é uma imagem de melhor qualidade que a original (FILHO; NETO., 1999).

1.3.3 Segmentação

Consiste em subdividir a imagem em regiões para facilitar a interpretação da mesma. A subdivisão é feita através do agrupamento de pixels utilizando características (features) comuns. Estas propriedades podem ser cores, intensidade, textura ou continuidade (ANJO, 2013).

A aplicação de filtros é umas das principais técnicas para extração de características de uma imagem sendo de grande importância segmentar uma região de interesse, o que reduz consideravelmente o tempo de processamento (PAVAN; CAZHURRIRO; MODESTO., 2010).

1.3.4 Extração de características

Nesta etapa o objetivo é extrair características das imagens resultantes da segmentação através de descritores que possam representar cada digito e diferenciar os dígitos

parecidos. Estes descritores devem ser representados por uma estrutura de dados adequada ao algoritmo de reconhecimento (FILHO; NETO., 1999).

1.3.5 Reconhecimento e interpretação

A classificação é processo de extração de informações em objetos para reconhecer padrões e objetos. Associa cada pixel da imagem ao um "*rótulo*" descrevendo um objeto real. Na classificação são extraídas da imagem informações mais convenientes à interpretação automática.

Em geral, refere-se ao agrupamento de dados em conjuntos similares. Esta informação é diversas vezes usadas em passos de analises para qualquer sistema de processamento de sinal/dados. A classificação de imagem é similar a classificação de dados em geral, mas pode variar dependendo da aplicação em que é utilizada (QIDWAI; CHEN, 2009).

O reconhecimento é o processo de atribuição de um rótulo a um objeto tendo como base suas características. A interpretação consiste em atribuir um significado a um conjunto de objetos (FILHO; NETO., 1999).

1.4 Trabalhos relacionados

Existem várias propostas para resolver o problema de reconhecimento de gestos. Podemos dividir estes trabalhos em dois grupos: um que usa sensores acoplados ao corpo e outro que usa visão computacional. Os que envolvem sensores acoplados ao corpo (exemplo as luvas) tendem a facilitar o processo de digitalização e resultados mais confiáveis, mas gerando desconforto ao usuário. Ja os de visão computacional só necessitam de uma câmera, sendo a sua desvantagem a variação de iluminação (MONTEIRO et al., 2016).

Para reconhecimento de gestos (PAVAN; CAZHURRIRO; MODESTO., 2010) utiliza uma webcam para capturar imagens dinâmicas e extrair as características da mão segmentada. Tendo como objetivo a criação de letras ou sinais suficientes para criação de palavras simples.

(SOARES; RAIA., 2014) apresenta um dispositivo na forma de vestuário equipado com um sensor de profundidade Kinect da Microsoft para auxiliar pessoas com limitações visuais. No projeto foi colocado um indivíduo num espaço fechado com vários obstáculos, durante o deslocamento o indivíduo recebia informações constantes, trazidas por meio de transdutores vibratórios embutidos na vestimenta. O projeto teve problemas na identificação de objetos com superfícies altamente reflexiva ou polida.

Em diversas áreas tem sido aplicado redes neurais. (ALVARENGA; CORREA; OSÓRIO., 2012) propõem um sistema para o aprendizado e classificação de gestos com a utilização do sensor de profundidade Kinect sendo executado em tempo real. Sendo

desenvolvido um programa supervisor em Python. O programa analisa a posição da mão direita, armazena pontos e deslocamentos em listas separadas. O programa foi desenvolvido para reconhecer três gestos: *Circle* ; *Come here* e *good bye*.

Um abordagem para classificação de um conjunto de nove sinais é proposta por (MENDONÇA., 2013) (entregar, pegar, abrir, olhar, empurrar, fechar, falar, puxar, trabalhar). Na captura é usado o Kinect e a biblioteca OpenNI. No processo de segmentação utiliza-se a informação do *skeleton* do Kinect para definir um quadro em torno da mão e segmentá-lo da imagem. Para descartar a informação do fundo é utilizada a imagem de profundidade.

Um sistema para reconhecimento automático das 26 posturas estáticas do alfabeto da língua brasileira de sinais (LIBRAS) e da língua de sinais americana (ASL) é proposto por (JUNIOR., 2014). Utiliza um sensor de RGB-D na aquisição de dados e de posse das imagens de profundidade aplica a combinação da estratégia de Casamento de Modelos, com o algoritmo *Interative Closest Point* (ICP) na fase de reconhecimento. O trabalho apresentou um desempenho máximo de 99,04 % de taxa de acerto no reconhecimento na ASL e 99,62 para a LIBRAS.

A criação de uma base de dados (em expansão) contendo 24 palavras em LIBRAS executadas por vários voluntários, e com planos de fundo diferentes é proposta por (MONTEIRO et al., 2016). A extração de características é baseada em subamostragens de imagens residuais. O trabalho apresentou um taxa de acerto média de 75%.

Tabela 1 – Equipamentos e gestos usados na literatura

Autor	**Dispositivo de captura**	**Gestos/Letras**
Pavan (2010)	Webcam	Letras A, B, C e D
Soares (2014)	Kinect 360 + vestuário	Sensores no corpo e kinect para identificar obstáculos em ambiente.
Alvarenga (2012)	Kinect 360	Gestos (Circle, Come here, Goodbye)
Mendonça (2013)	Kinect 360	Criado videos (Entregar, pegar, abrir, olhar, empurrar, fechar, falar, puxar, trabalhar)
Juarez (2014)	Kinect 360	26 posturas estáticas (LETRAS A a Z em ASL e LIBRAS)
Monteiro (2016)	Kinect 360	Gestos (Andaime, depressão, fantasma, martelo, nacionalidade, paciência, parede, neve, redação, relâmpago, trampolim, vapor)

Fonte: Autor

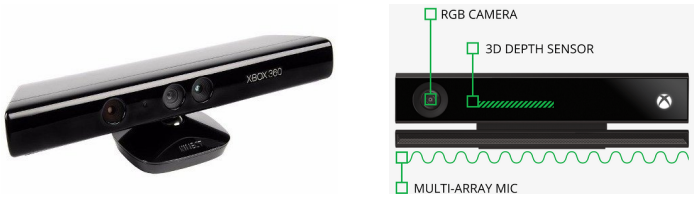
2 REFERENCIAL TEÓRICO

2.1 Sensor Kinect e Kinect SDK

Na captura da imagem foi utilizado o sensor Kinect One apresentado na Figura 2 que é formado por um laser infravermelho e um sensor CMOS monocromático, que capta os dados 3D. Uma das características que este sensor de profundidade apresenta é que ele não é sensível a variação da luz. Ele é capaz de capturar imagens tanto em ambiente bem iluminado quanto em ambiente escuro (MENDONÇA., 2013).

A resolução da câmera de profundidade é de 512x424, da câmera em RBG é de 1920x1080 (16:9), uma razão de 30 frames por segundo e resolução do microfone de 48kHz.

O Kinect (360 e One) também possui uma câmera de cores (RGB), um sensor de profundidade que usa infravermelho e mapea ambiente e 3D, microfones, uma base com motor para alterar seu angulo de visão e interface USB para conexão com videogame e também com o computador. A robótica vem fazendo uso deste dispositivo, devido a sua capacidade de percepção espacial.



(a) Kinect 360 (b) Kinect one

Figura 2 – Sensor Kinect

O Kinect possui uma tecnologia desenvolvida pela empresa Rare, subsidiária da Microsoft Games Studios. Seu sensor de profundidade é produzido pela companhia israelense PrimeSense, que batizou sua tecnologia de escaneamento 3D de Light Coding (MENDONÇA., 2013).

A Microsoft disponibiliza o aplicativo KinectSDK versão 1.8 para utilização do Kinect e Windows. Com este aplicativo pode-se manipular as imagens geradas pelo dispositivo, capturar áudios através de microfones embutidos e também utilizar algorítmos de segmentação de usuários e modelagem do corpo humano (ANJO, 2013).

O pacote oferece acesso as fontes de dados captadas pelos sensores: de profundidade, câmera RGB e 4 microfones do sistema de captura de áudio. Permite ainda acesso ao rastreio do esqueleto (CORREIA, 2013).

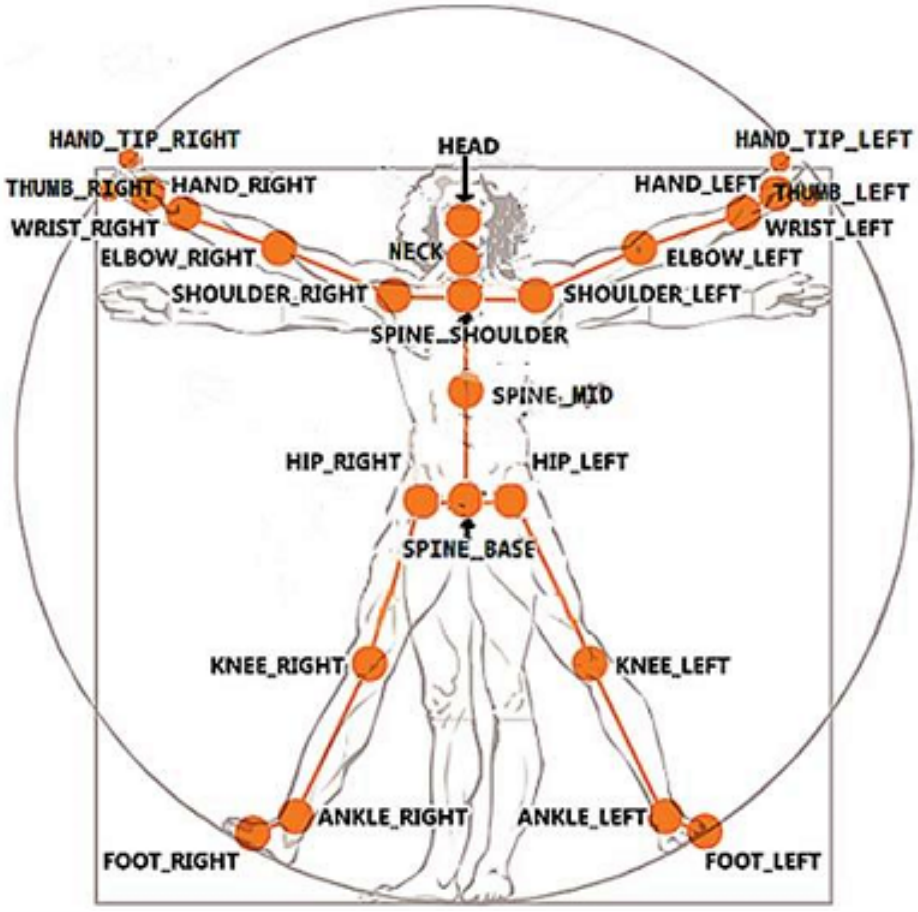
Tabela 2 – Comparação entre o kinect V1 e V2

	V1 (xbox 360)	V2 (xbox one)
Alcance do sensor de profundidade	0.4 - 4.0 m	0.4 - 4.0 m
Resolução do canal colorido	640x480	1920x1080
Resolução do canal de profundidade	320x240	512x424
Canal infravermelho	None	512x424
Tipo de luz	Light coding	Tof
Canal de áudio	4-mic array 16 kHz	4-mic array 48 kHz
USB	2.0	3.0
Juntas	20	25
Campo de visão	57^0 H 43^0 V	70^0 H 60^0 V

Fonte: (MathWorks, 2017)

Na Tabela 2 podemos observar a diferença entre os dois sensores Kinect. A primeira é na qualidade das imagens adquiridas, na imagem RGB do One a resolução é full HD. Outra diferença importante está na quantidade de articulações que o novo sensor pode capturar. No sensor Kinect 360 conseguimos informações sobre 20 articulações, enquanto que no Kinect One temos 25 articulações do corpo do usuário, conforme a Figura 3.

Figura 3 – Pontos de articulação do corpo obtido pelo Kinect ONE



Fonte: (Microsoft,2017)

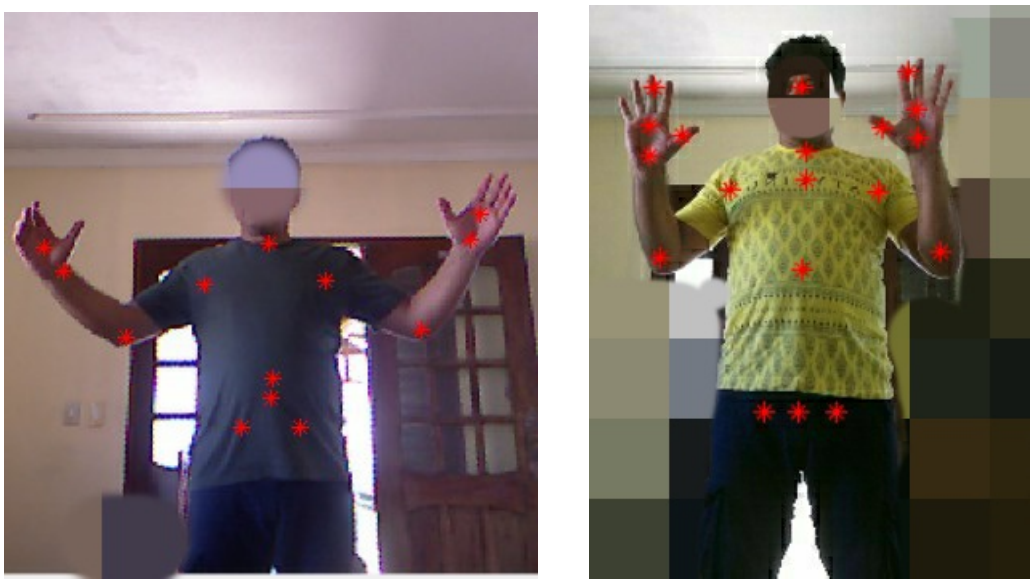
Cada articulação do corpo é representada por um ponto indicado na Tabela 3. Nesta tabela foram apresentados as 25 articulações do Kinect One. Para o sensor Kinect 360 até a articulação de numero 20 são coincidentes com o One.

Tabela 3 – Pontos das articulações do corpo

SpineBase	1	Knee_Left	14
SpineMid	2	Ankle_Left	15
Neckr	3	Foot_Left	16
Head	4	Hip_Right	17
Shoulder_Left	5	Knee_Right	18
Elbow_Left	6	Ankle_Right	19
Wrist_Left	7	Foot_Right	20
Hand_Left	8	SpineShoulder	21
Shoulder_Right	9	HandTip_Left	22
Elbow_Right	10	Thumb_Left	23
Wrist_Right	11	HandTip_Right	24
Hand_Right	12	Thumb_Right	25
Hip_Left	13		

Fonte: (MathWorks, 2017)

As 5 articulações a mais que o sensor Kinect One trás em seu pacote é exibida na Figura 4b. Podemos destacar os pontos 22 e 24 que representam as pontas do dedos indicadores das mãos, os pontos 23 e 25 que representam os polegares das mãos e o ponto 21 que representa o centro dos ombros. Estes pontos são importantes no uso do vídeo game, onde o usuário muda o comando se a mão estiver aberta ou fechada.



(a) Kinect 360 (b) Kinect One

Figura 4 – Pontos de articulação dos sensores Kinect

Numa imagem de profundidade podemos obter as informações tridimensionais precisas. A câmera tradicional captura uma imagem como elas se parecem (ver Figura 5), uma câmera de profundidade captura como elas são (ver Figura 6) (SOARES; RAIA., 2014).

Figura 5 – Imagem da letra C sensor RGB

Fonte: Autor

A imagem de profundidade facilita na modelagem de partes do corpo humano. Sendo sua principal vantagem na modelagem é quanto à sobreposição de partes do corpo. No caso de braço e corpo seria facilmente detectada e segmentada com a característica de profundidade.

Figura 6 – Imagem da letra C sensor de profundidade

Fonte: Autor

O Kinect é capaz de reconhecer um corpo humano acompanhando seus movimentos, pode também estimar algumas juntas do corpo. Este recurso é utilizado em inúmeras

outras aplicações, pois é uma forma rápida de rastrear o usuário. A Figura 7 mostra uma captura utilizando o Kinect.

Figura 7 – Skeleton do corpo utilizando kinect

Fonte: Autor

2.2 Seleção da região de interesse - ROI (Region of Interest)

A segmentação é um processo de agrupamento de pixels que possuem características semelhantes. Trata-se da decomposição da imagem em regiões discretas. Neste processo utiliza-se algoritmos de segmentação de imagens que ao definir regiões homogêneas na imagem, prepara para a classificação posterior (MENESES; ALMEIDA, 2012).

Nas pesquisas relacionadas à visão computacional e a reconhecimento de gestos são utilizados técnicas de aplicação de filtros a uma determinado segmento da imagem capturada (PAVAN; CAZHURRIRO; MODESTO., 2010).

Na imagem de profundidade o sensor irá captar informações sobre toda a cena. Para uma processamento mais rápido e eficiente, é necessário reduzir a quantidade de informações disponível para as informações que são mais relevantes ao nosso estudo. Este etapa chamada de pré-processamento (CORREIA, 2013).

Existe várias formas de segmentação, como por exemplo: métodos baseados em histogramas; Edge Detection; segmentação através de grafos; segmentação de regiões através da utilização de redes neurais como Multi-Layer Percepton (MLP), entre outras (ANJO, 2013).

São vários os algoritmos de segmentação e são baseados, em geral, em duas propriedades básicas de valores de cinza: descontinuidade e similaridade. Na descontinuidade a abordagem é detecção de pontos isolados, detecção de linhas e bordas na imagem, ela segrega a imagem conforme mudanças abruptas nos níveis de cinza. Já a similaridade baseia-se nos métodos de limiarização, crescimento de regiões e divisão.

Fica a cargo do usuário definir o limiar de similaridade e o tamanho mínimo dos polígonos que serão gerados. Esse processo leva a tentativa e erros até encontramos o

limiar desejado.

2.2.1 Histograma

As funções de transformação de intensidade com base nas informações extraídas dos histogramas de intesidade da imagem desempenham um papel básico no processamento de imagens. (GONZALES; WOODS; EDDINS., 2004)

O histograma de uma imagem digital com k níveis de cinza é uma função discreta dada pela equação 2.1.

$$p(k) = \frac{n_k}{n} \tag{2.1}$$

sendo:

k = nível de cinza, podendo variar de 0 (preto) a 255 (branco);

n_k = número de pixels na imagem com nível de cinza k;

n = número total de pixels da imagem;

p_k = estimativa de probabilidade de ocorrência do nível de cinza k.

Na técnica do histograma de uma imagem obtemos um conjunto de números indicando o percentual de pixels naquela imagem que apresenta um determinado nível de cinza. Os valores são representados por um gráfico de barras (FILHO; NETO., 1999).

Embora o histograma não diga nada a respeito do conteúdo da imagem, a informação extraída é muito útil para seu processamento. O histograma tem um papel importante na etapa da segmentação, os picos de intensidade correspondem as fases presentes, permitindo a separação e/ou quantificação de cada uma delas.

Na Figura 8 temos um conjunto de informações desnecessárias para nosso trabalho, sendo de interesse somente a região do quadro em vermelho.

Figura 8 – Imagem da letra B

Fonte: Autor

Quando necessitamos separar o fundo do objeto podemos usar a técnica de limiarização. A forma mais simples consiste em fazer uma bipartição do histograma, convertendo os pixels cujo valor são maiores ou iguais ao valor estabelecido (T) em branco e os demais em preto (FILHO; NETO., 1999).

A Figura 9 nos apresenta somente a região de interesse o que diminui o custo computacional.

Figura 9 – Imagem da letra B segmentada

Fonte: Autor

2.3 Extração de características

A extração de características é uma tarefa fundamental para o processo de reconhecimento e depende fortemente do objeto. A quantidade de informações que se pode extrair de uma imagem é que garante a capacidade de se reconhecer um objeto.

O nosso trabalho tem como base realizar o rastreio da mão. Após a mão ser detectada e segmentada, iniciamos a extração das características. A detecção do contorno da mão pode ser obtida tendo a imagem segmentada de uma forma binária, pode-se

considerar que o primeiro pixel com valor 1 encontrado num varrimento lateral faz parte do contorno do objeto (CORREIA, 2013).

2.3.1 Filtros

Uma imagem é composta de uma estrutura espacial com regiões de altas e baixas frequências, matematicamente pode ser escrita como:

$$f(x, y) = PB(x, y) + PA(x, y) \tag{2.2}$$

sendo:

$PA =$ passa-baixa;

$PB =$ passa-alta;

A decomposição da imagem em componentes de baixa e altas frequências de brilho é a base para a filtragem espacial. Na maioria dos filtros se utilizam um operador de convolução discreta, que vai operar dois elementos distintos, a imagem e o filtro (MENESES; ALMEIDA, 2012).

Na maioria dos filtros é utilizada uma janela deslizante. Dois procedimentos são utilizados para realizar a filtragem:

- O primeiro define uma máscara de arranjo de uma pequena matriz contendo coeficientes e pesos. A matriz de pesos é chamada de kernel de convolução, normalmente se usa um numero impar de pixels para que seja mantida a simetria em relação ao pixel central.
- A máscara movida sobre a imagem, linha por linha, coluna por coluna, e os valores dos pixels de cada área da imagem sob o filtro são multiplicados pelos correspondentes pesos dos pixels do filtro. A média da soma deste produto será o novo valor de brilho do pixel situado na posição central da área da imagem sob o filtro. Este valor é salvo e o processo continua.

O processo é repetido varrendo-se toda a imagem pixel a pixel.

Os métodos de filtragem pode sem classificados em duas categorias: Técnicas de filtragem espacial e as no domínio da frequência. As que trabalham no domínio espacial operam com a matriz de pixel que é a imagem digitalizada. Os métodos no domínio da frequência se baseiam na modificação da transformada de Fourrier (FILHO; NETO., 1999).

As funções de processamento de imagens no domínio espacial podem ser expressas por:

$$g(x, y) = T[f(x, y)] \tag{2.3}$$

sendo: $g(x, y)$ é a imagem processada, $f(x, y)$ é a imagem original e T é um operador em f, definido em uma certa vizinhança de (x, y).

Nas técnicas de filtragem no domínio da frequência a base matemática é o teorema da convolução. Seja $g(x, y)$ a imagem formada pela convolução (denotada pelo simbolo *) da imagem $f(x, y)$ com o operador linear $h(x, y)$, ou seja,

$$g(x, y) = f(x, y) * h(x, y) \tag{2.4}$$

Então pelo teorema da convolução, a seguinte relação no domínio da frequência também é válida:

$$G(u, v) = F(u, v)H(u, v) \tag{2.5}$$

No domínio espacial são utilizadas máscaras que são conhecidas como filtros espaciais. Os filtros podem ser: passa-baixa (quando atenuam ou eliminam componentes de alta frequência); passa-alta (quando atenuam ou eliminam componentes de baixa frequência) ou passa-faixa (capazes de remover ou atenuar componentes acima de sua frequência de corte superior e abaixo da sua frequência de corte inferior) (FILHO; NETO., 1999).

Os filtros passa-altas realçam as bordas e regiões de alto contraste da imagem, sendo indicado para extração dos contornos da mão.

O laplaciano de uma função de duas variáveis $f(x, y)$ é definida como:

$$\bigtriangledown^2 f(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \tag{2.6}$$

O laplaciano é um operador útil no processo de detecção de bordas, como pode ser visto na Figura 10.

2.4 Classificação e reconhecimento

Na etapa de classificação é importante a utilização de imagem de referência já classificada, utilizando qualquer outro método, supervisionado ou não. A imagem de referência tem a finalidade de designar a classe majoritária em cada segmento (MENESES; ALMEIDA, 2012).

Existe um dicionário online disponibilizado pelo Instituto Nacional de Eduação de Surdos (INES) que contem 3853 sinais/itens lexicais. Os verbetes são apresentados em

Figura 10 – Imagem da letra B filtro laplaciano

Fonte: Autor

uma sequência de vídeo com resolução de 240 x 180 pixels, sendo executado para uma mulher num ambiente controlado (MONTEIRO et al., 2016). As imagens (ver Figura 11) fornecidas pelo INES serão utilizadas para garantir confiabilidade ao trabalho.

Figura 11 – Imagem da pessoa realizando gesto da letra B

Fonte: (INES, 2005)

Os classificadores podem ser divididos em: supervisionados que são obtidos a partir de exemplos conhecidos; e não supervisionados, onde não tem um conjunto de exemplos de treinamento, o algoritmo aprende a obter agrupamentos ("*clustering*") dos vetores de entrada seguindo algum critério de similaridade. Este tipo de aprendizado busca encontrar tendências e ou padrões visando um melhor entendimento dos dados experimentais. Os não supervisionados destacam-se: o distância mínima, máxima verossimilhança; árvores de decisão; redes neurais e *Support Vector Machines* (SVM).

2.4.1 Distância mínima

O método da distância mínima calcula a distância entre o vetor de medida e o pixel candidato. O método utiliza a distância Euclidiana. Cada pixel se atribuído a uma classe através da analise de similaridade de distância Euclidiana, que é dada por:

$$D(x,n) = \sqrt{(x_i - m_i)^2} \tag{2.7}$$

sendo:

x_i = pixel candidato;

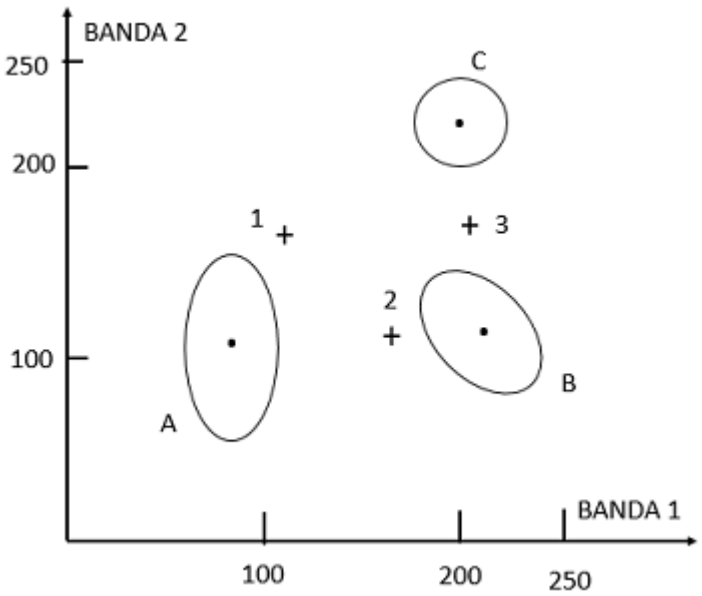
m_i = média das classes;

n = numero de bandas.

Esse método atribui a cada pixel desconhecido a classe cuja média é mais próxima a ele. Este método considera a questão da proximidade entre classes com base em dados estatísticos. Podemos ver na Figura 12 o pixel 1 situa-se mais próximo da classe A, ja o pixel 2 mais próximo da classe B. A desvantagem deste método é quando um pixel se encontra na mesma distância das médias de duas classes, necessitando outros métodos.

A vantagem deste método é que todos os pixels serão atribuídos a uma classe, não existindo pixels não classificado. As desvantagens são: pixel que não deveriam ser classificados; o método não avalia a variabilidade espacial.

Figura 12 – Espaço de atributos das classes A, B e C



Fonte: Adaptado (BELUCO., 2002)

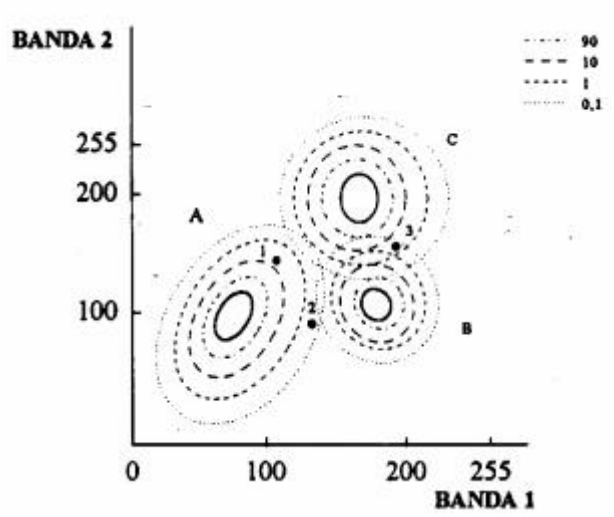
2.4.2 Máxima verossimilhança

É o método mais utilizado de classificação de imagens de sensoriamento remoto. Este método considera a ponderação das distâncias entre as médias através de dados estatísticos. É um classificador mais eficiente porque utiliza as classes de treinamento para estimar a forma de distribuição dos pixels de cada classe, como também a localização do centro de cada classe (MENESES; ALMEIDA, 2012).

A classificação é feita a partir da probabilidade de cada pixels pertencer a uma determinada classe, ou seja, será atribuído um pixels pertencer a uma classe A, se a probabilidade dele for maior do que pertencer a qualquer outra classe.

Na Figura 13 podemos observar que o pixel 2 se encontra mais próximo da media da classe B, mas esta dentro das linhas de contorno da média da classe A. Então o pixel 2 tem maior probabilidade de pertencer a classe A.

Figura 13 – Representação das curvas de probabilidade de ocorrência das classes A, B e C



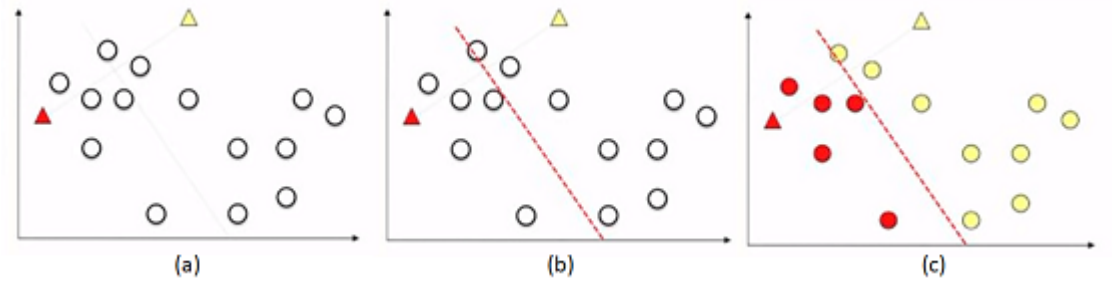
Fonte: Adaptado (ALMEIDA, 2013)

2.4.3 Algoritmo K-Médias ou K-Means

O algoritmo é uma técnica interativa para particionar um conjunto de dados em grupos separados, onde o valor de k, deve ser predeterminado. Se baseia na minimização de uma média custo, a distância interna entre os padrões de um agrupamento. Essa minimização do custo encontra um mínimo local da função objetivo, que dependerá do ponto inicial do algoritmo, (Prado and Monterio, 2008)

Primeiramente são atribuídos centróides aleatórios conforme pode ser visto na figura 14a. Entre este dois centróides traça-se uma linha, visto em 14b. Com esta linha traçada se divide as classes em dois grupos, mostrado em 14c.

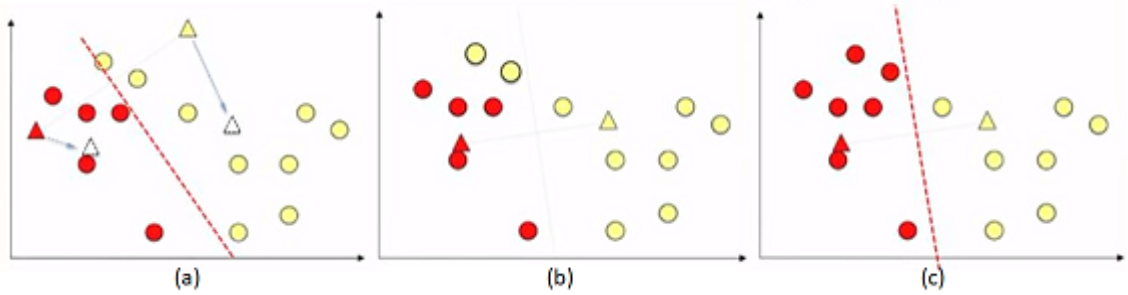
Figura 14 – Primeira interação do algoritmo



Fonte: Adaptado (LAVRENKO, 2014).

Então recalcula-se os novos centroides deste dois grupos 15a, os novos centroides encontrados traça-se novamente uma linha. Esta nova linha divide as classes, observa-se que classes que perteciam a um grupo agora estão pertencendo ao outro, ver 15c.

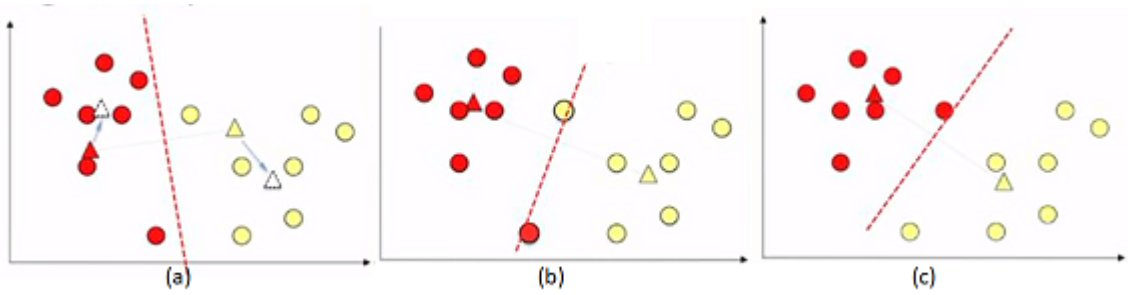
Figura 15 – Segunda interação do algoritmo



Fonte: Adaptado (LAVRENKO, 2014).

Repete-se os passos anteriores com a busca por novos centroides ate que as variações destes valores sejam minimas. Não havendo mais variações significativas dos centroides as classes são divididas.

Figura 16 – Terceira interação do algoritmo



Fonte: Adaptado (LAVRENKO, 2014).

O comportamento do algoritmo K-Means apresenta vantagens com relação a simplicidade e eficiência. É rápido para cálculos simples. O erro quadrático total (TSE) decresce (ou converge) a cada interação.

2.4.4 Support Vector Machines (SVM)

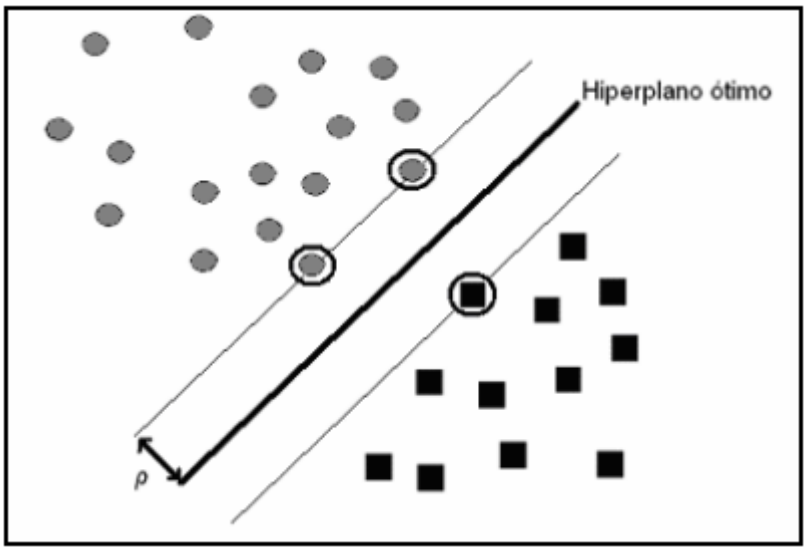
Os resultados obtidos com SVM tem sido tem sido superiores as outras técnicas ja consagradas. As SVMs tem sido utilizadas em reconhecimentos de faces, caracteres manuscritos, impressões digitais, etc (CHACON., 2012).

A SVM trata o problema de classificação utilizando uma função de mapeamento (linear ou não) que transforma os dados do espaço de características original para outro espaço, geralmente, de maior dimensão, tornando assim o problema separável.

Os algoritmos de aprendizagem de máquinas (SVM) determina limites de decisão produzindo uma separação ótima entre as classes por meio da minimização dos erros. Sendo uma técnica computacional de aprendizado para reconhecimento de padrões (NASCIMENTO et al., 2009).

Assumindo que duas classes de amostras de treinamento são linearmente separáveis, a função de decisão é aquela em que a distância das amostras é maximizada. A função que maximiza esta separação é denominada ótima (ver Figura 17).

Figura 17 – Hiperplano ótimo separando as duas classes



Fonte:(ANDREOLA, 2009)

O hiperplano pode ser descrito pela equação:

$$y(x) = w^T x + b \tag{2.8}$$

onde todos os vetores x que validam esta equação estão no plano e $\vec{w}$ e b devem ser determinados.

Adotando a seguinte notação para os vetores da classe 1 (Ver Figura 18).

$$w^T x_1 - b = +1 \tag{2.9}$$

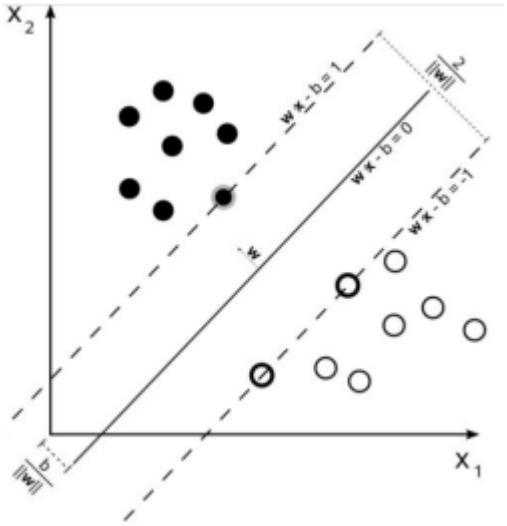
a para os vetores da classe 2

$$w^T x_2 - b = -1 \tag{2.10}$$

estas pode ser rescritas por:

$$y_i.(w^T x_i - b) \geq 1 \quad \textit{para todos os } i \tag{2.11}$$

Figura 18 – Hiperplano de separação com os vetores de suporte das duas classes em destaque. Determinação do plano ótimo de separação das duas classes.



Fonte: (CHACON., 2012)

sendo a equação do plano:

$$w^T x - b = 0 \tag{2.12}$$

Os vetores característicos x são linearmente separáveis, com isto podemos usar estes dois hiperplanos na margem das classes para tentar maximizar a distância entre eles. vendo a Figura 18 e por meio de uma analise geométrica podemos concluir:

w é um vetor normal ao hiperplano, $w = w_1, w_2, w_3 \ldots, w_n$

e $\|w\|$ é a norma euclidiana de $w = \sqrt{w_1^2, w_2^2, w_3^2 \ldots w_n^2}$

e $|b|/\|w\|$ é a distância do hiperplano até a origem.

A distância d de um ponto x qualquer ao hiperplano é dada por:

$$d = \frac{(w^T x + b)}{\|w\|} \tag{2.13}$$

Logo a distância entre os dois planos é: $2/\|w\|$. O problema de minimização desta distância consiste em maximizar o $\|w\|$. A solução não é muito simples porque envolve a raiz quadrada para resolver a norma. Sendo possível alterar a equação para:

$$\frac{1}{2}\|w\|^2 \tag{2.14}$$

Para o treinamento do classificador se faz necessário um conjunto de imagens que servirão como referencia, em diversas condições e posições diferentes. Este conjunto são as amostras positivas, também se faz necessário um conjunto de imagens aleatórias onde

nossos objetos de estudo não estejam. Este conjunto são as amostras negativas (PAVAN; CAZHURRIRO; MODESTO., 2010).

2.4.5 Redes Neurais Artificiais

São modelos matemáticos que se inspiram nas redes neurais biológicas, tendo capacidade computacional de adquirida por meio de adaptação, apresentando propriedades de aprendizado e generalização.

A rede neural artificial (RNA) é baseada neurônio biológico. Para cada neurônio foi desenvolvido um modelo matemático, uma combinação destes neurônios artificiais constitui uma rede.

Os neurônios biológicos são células capazes de processar, receber e enviar sinais elétricos, já os neurônios artificiais são representações numéricas destas células que computam funções matemáticas, buscando de certo modo reproduzir os potenciais elétricos das células e sua propagação (ALVARENGA; CORREA; OSÓRIO., 2012).

As diferentes formas de conexões dos neurônios fazem a diferenciação entre os tipos de redes neurais e suas diferentes aplicações.

2.4.5.1 Arquitetura da Rede Neural

A arquitetura de uma rede neural são de diversas formas, estando ligada diretamente ao algorítmo de aprendizagem, usado para treinar a rede. Os itens que compõem a estrutura de uma rede neural são:

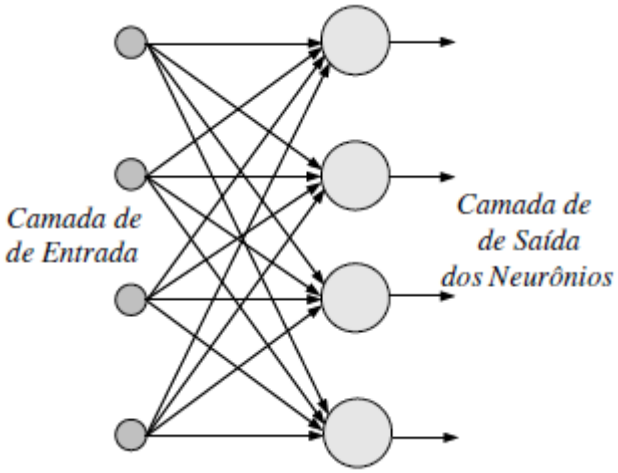
- Camadas intermediárias;
- Quantidade de neurônios;
- Função de transferência;
- Algoritmo de aprendizado.

A maneira que os neurônios de uma rede são estruturados esta intimamente ligado ao algorítmo de aprendizagem. Pode-se falar de algorítmos de aprendizagem como sendo estruturados. Podemos classificar três tipos de arquitetura diferentes.

2.4.5.1.1 Redes Alimentadas Adiante com Camada Única

Em uma rede neural de camadas, os neurônios estão organizados na forma de camadas. O termo "camada única"se refere à camada de saída de nós computacionais (ver Figura 19).

Figura 19 – Rede alimentadas com camada única



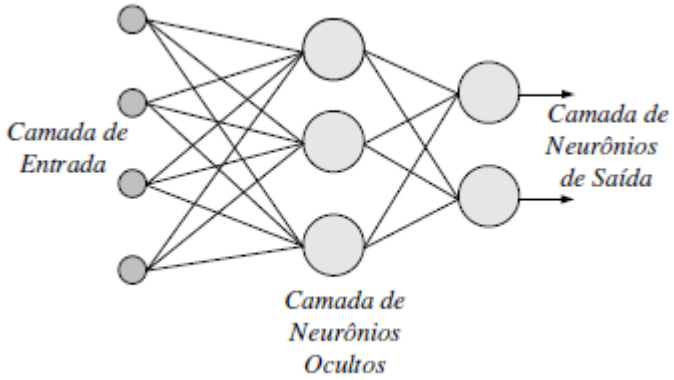
Fonte: (MATSUNAGA, 2012)

2.4.5.1.2 Redes Alimentadas Diretamente com Múltiplas Camadas

Nesse tipo de arquitetura temos a presença de uma ou mais camadas intermediárias (ou ocultas), cujos nós computacionais são chamadas de neurônios ocultos ou unidades ocultas. A função dos neurônios ocultos é intervi entre a entrada e a saída de maneira útil. Adicionando uma ou mais camadas ocultas, tornamos a rede capaz de extrair estatísticas de ordem elevada.

A Figura 20 mostra uma rede de 2 camadas com 4 entradas e 2 saídas.

Figura 20 – Rede alimentadas com múltiplas camada



Fonte: (MATSUNAGA, 2012)

2.4.5.1.3 Regra de Aprendizado por Retropropagação (Back-propagation)

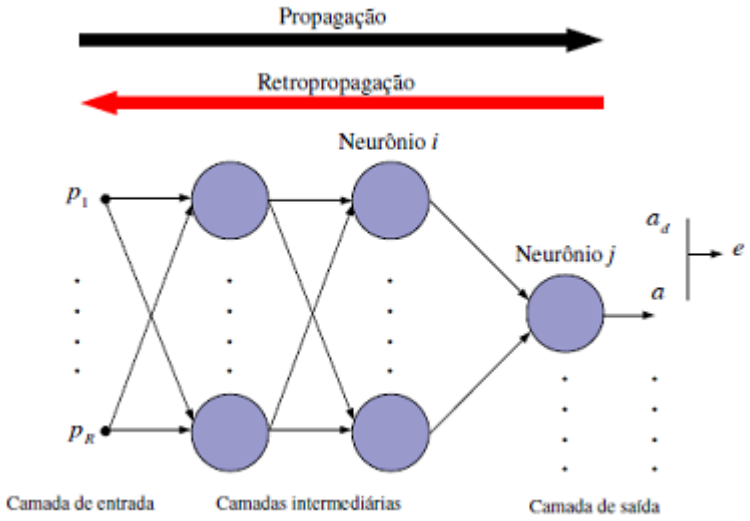
O algorítmo back-propagation de forma interativa procura achar a mínima diferença entre as saídas desejadas e as saídas obtidas pela rede neural artificial, com o mínimo de erro. Dessa maneira, ajustando os pesos entre as camadas através da retropropagação do erro encontrado em cada interação.

Esse método é do tipo treinamento supervisionado, onde a rede é analisada em dois casos: na sua propagação (camada por camada) e na sua retropropagação (análise contrária a propagação), Backpropagation.

Um padrão de entrada é aplicado aos elementos da primeira camada da rede, que é propagado por cada uma das outras camadas até que a saída seja gerada a. Esta é comparada com a saída a_d (gerando um sinal de erro e para cada elemento de saída). O sinal de erro é tão retropropagado da camada de saída para cada elemento da camada intermediária anterior que contribui diretamente para a formação de saída.

A ilustração do algorítmo Backpropagation pode ser ilustrado na Figura 21.

Figura 21 – Ilustração do algorítmo Backpropagation



Fonte: (MATSUNAGA, 2012)

2.4.6 HOG - Histograma de gradiente orientado

É um tipo de descritor de recursos usado em processamento de imagem com a finalidade de detecção de objetos. A técnica conta as ocorrências de orientação gradiente localizadas em uma imagem.

Calculando o HOG,

- Divida a imagem em blocos de tamanhos iguais;
- Cada bloco deve ser dividido em células de tamanhos iguais;
- Para cada célula obtenha o histograma de orientações;

  Obter o magnitude:

$$S = \left(\sqrt{S_x^2 + S_y^2}\right) \tag{2.15}$$

Figura 22 – Blocos do HOG

Fonte: (BRAZ, 2018)

Obter orientação

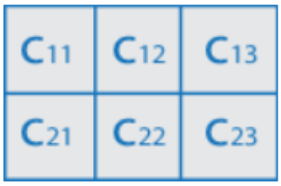
$$\Theta = arctan\left(\frac{S_y}{S_x}\right) \tag{2.16}$$

Exemplo no matlab.

Arranjo de histogramas em vetores de característica de HOG

A Figura 23 mostra uma imagem com seis células.

Figura 23 – Células de uma imagem



Fonte: (MATHWORKS, 2013)

Se você definir o BlockSize [2 2], seria o tamanho de cada porco bloquear, 2 por 2 células. O tamanho das células são em pixels. Você pode definir isso com a propriedade CellSize

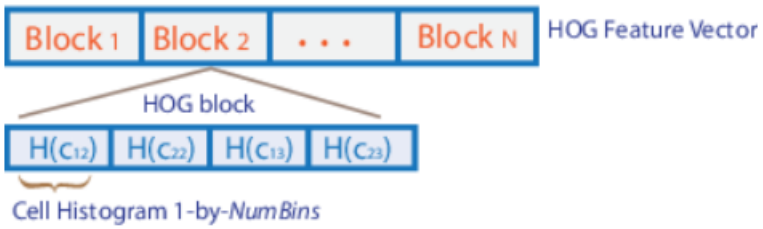
Figura 24 – Bloco 2 x 2

Fonte: (MATHWORKS, 2013)

O vetor de característica de HOG é organizado por blocos de HOG. O histograma de célula, H (CYX), é 1-por-NumBins.

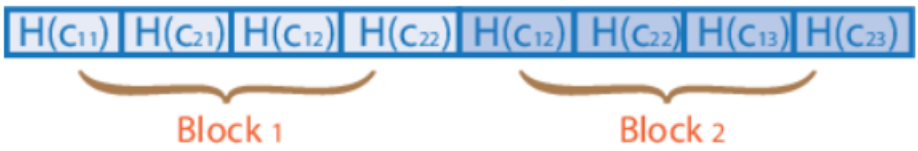
A Figura 26 mostra o vetor de característica de HOG com uma sobreposição de 1 por 1 célula entre blocos.

Figura 25 – Vetor de características HOG



Fonte: (MATHWORKS, 2013)

Figura 26 – Vetor de características HOG com sobreposição



Fonte: (MATHWORKS, 2013)

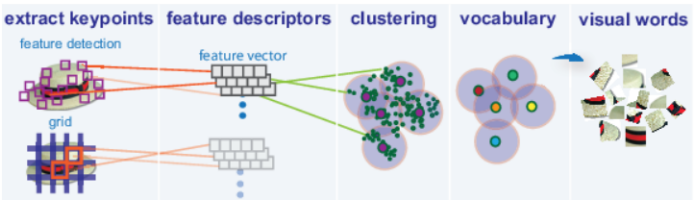
2.4.7 Bag of Visual Words

A abordagem Bag-of-Visual-Words (BoVW) é uma técnica robusta usada em métodos de classificação de documentos, onde a ocorrência de cada palavra é usada como um recurso para treinar um classificador (RODRíGUEZ, 2014). Um dicionário visual é definido e cada característica local extraída da imagem é tratada como uma palavra visual desse dicionário (PEDROSA, 2015).

Para representar um imagem em BoVW pode ser definido pelas seguintes fases: Representação de características; geração de vocabulário; contagem da ocorrência (frequência) de cada palavra visual contida na imagem.

- Representação das características:
- Geração de vocabulário:
- Geração do histograma;

Figura 27 – Bag of Visual Words



Fonte: (MATHWORKS, 2019)

3 TRABALHO PROPOSTO

Tendo em vista a dificuldade em comunicação entre pessoas que só conseguem se comunicar através da linguagem de sinais (LIBRAS) e pessoas que não possuem domínio na língua, o presente trabalho traz uma alternativa com a utilização do sensor Kinect (utilizado em vídeo game Xbox) e o software Matlab.

3.1 Metodologia

Para realização deste projeto foi utilizado Kinect que possui um sensor de profundidade que não é sensível a variação da luz, facilitando a captura de imagens em ambiente bem iluminados como também em ambientes com variações.

A interface do Kinect com o computador foi realizada através do software Matlab R2018a e R2015a. O Kinect tem duas entradas de vídeos, sendo 'Kinect Color Sensor' responsável pela captura das imagens nas escalas RGB e o 'Kinect Depth Sensor' para captura da imagens de profundidade, suas resoluções foram apresentadas na Tabela 2.

Na aquisição foram utilizados: Um notebook com o software Matlab 2015a com o sensor Kinect 360 e outro notebook com o Matlab 2018a com o sensor Kinect ONE. Os sensores ficaram a uma altura de aproximadamente 0,9m, os atores ficaram a uma distância do sensor de 1,5m e com um fundo a uma distancia de 3,8m.

Figura 28 – Notebooks e sensores Kinect

Fonte: Autor

A coleta de imagens foi realizada nas cidades de São Luís e Fortaleza em dias da semana diferentes, com fundo variados. Cada ator ficava em frente aos sensores Kinects variando a rotação da mão num intervalo de tempo necessário para uma captura de pelo menos 100 imagens.

Para uma melhor diversidade no banco de dados os atores são variados em idade, sexo, cor e estaturas e são detalhados na Tabela 4. Essa diversidade foi aleatória de acordo com a disponibilidade de cada ator, não houve um estudo para escolha dos mesmos.

As imagens foram adquiridas no campo de visão de 70^0 horizontalmente e 60^0 verticalmente, num alcance de profundidade de 0,4m a 4,5m. Limites do sensor kinect.

Tabela 4 – Variações do atores

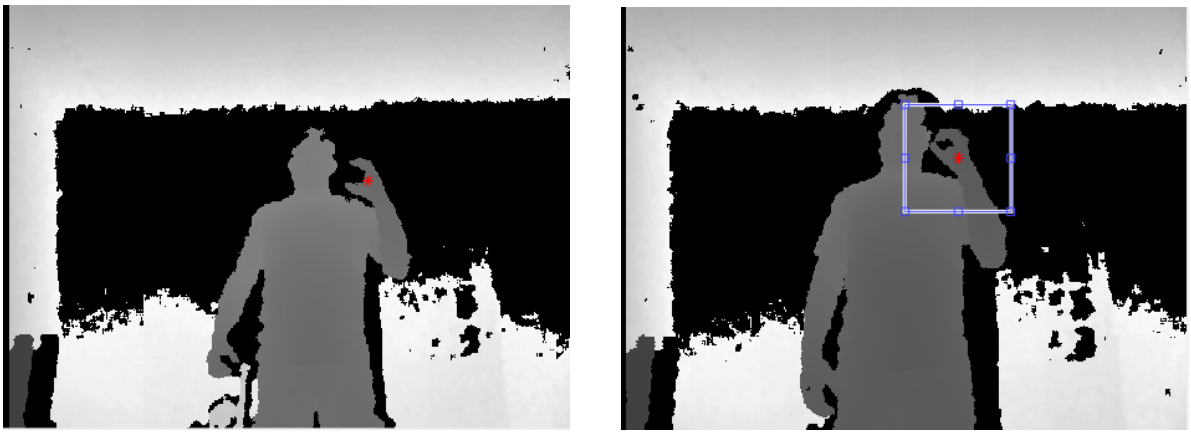
Ator	**Idade (Anos)**	**Altura (metros)**	**Sexo**	**Cor da pele**
1	19	1,60	Femenino	Parda
2	44	1,60	Masculino	Branca
3	46	1,78	Masculino	Parda
4	50	1,60	Femenino	Branca
5	25	1,55	Femenino	Branca
6	22	1,76	Masculino	Parda
7	27	1,76	Masculino	Negra

Fonte: Autor

No trabalho utilizamos comandos para criação de um vídeo de entrada no ambiente Matlab. Sendo que criamos dois objetos de entrada, uma para imagem em RGB e outra para a imagem de profundidade.

Foi utilizada a função "*Body*" do Matlab que fornece informações sobre a posição de 25 articulações do corpo do usuário. Cada articulação recebe um numero e para uma melhor visualização do movimento do corpo utilizamos a função 'SkeletonConnectionMap'. Para visualizarmos o movimento do corpo através do skeleton, vamos criar um vetor com vários outros vetores, sendo que cada vetor irá fazer a ligação de uma articulação para outra, desta maneira podemos ilustrar o corpo.

Dentre estas articulações nossos estudos focam nos movimentos e acompanhamento das mãos. Esta facilidade do função em distinguir as mãos dos restante do corpo em qualquer posição da imagem ajuda na determinação da região de interesse (ROI) diminuindo o tempo computacional.



(a) Ponto de articulação da mão direita (b) ROI da mão direita

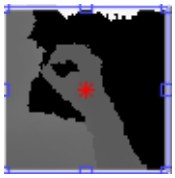
Figura 29 – Ponto da mão direita e ROI - Sensor 360 profundidade

O sensor kinect reconhece as articulações do corpo humano, para a articulação da mão temos um ponto conforme pode ser visto na 29a. Ao redor do ponto que rastrea a mão direita foi criado um retângulo de forma arbitrária de 101x101 (Kinect 360) visto

na Figura 29b e 161x161 (Kinect One). Os demais pontos foram omitidos pois não serão necessários nesta parte do trabalho.

Com a criação deste retângulo conseguimos definir nossa região de interesse e retirar somente a região que nos interessa, as ROI's. Ver figura 30.

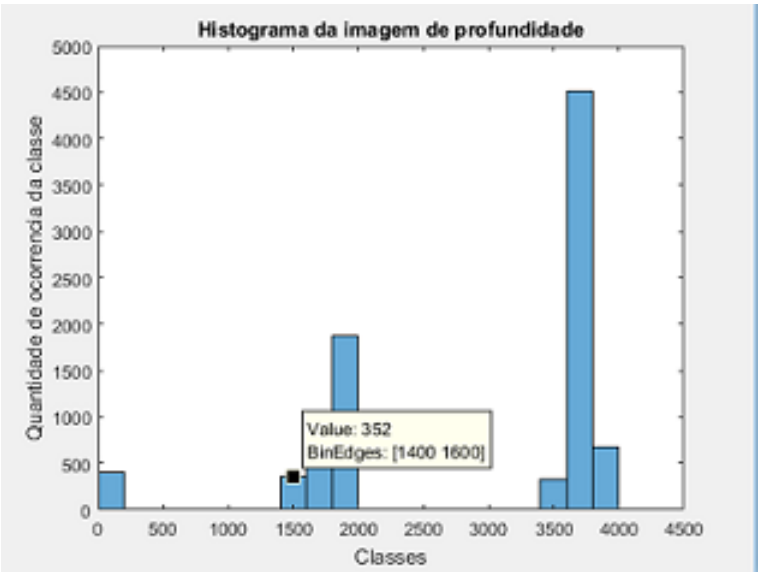
Figura 30 – Imagem recortada - Kinect 360 depth



Fonte: Autor

Na Figura 30 podemos observar que temos muitas informações desnecessárias, como por exemplo o fundo da imagem. No Matlab existem vários comandos que calculam o histograma das imagens, porém com a utilização destes comando perdemos os valores de profundidade. Para resolvermos este conflito criamos uma função que faz a leitura pixel a pixel para depois exibirmos o histograma conforme pode ser visto na Figura 31.

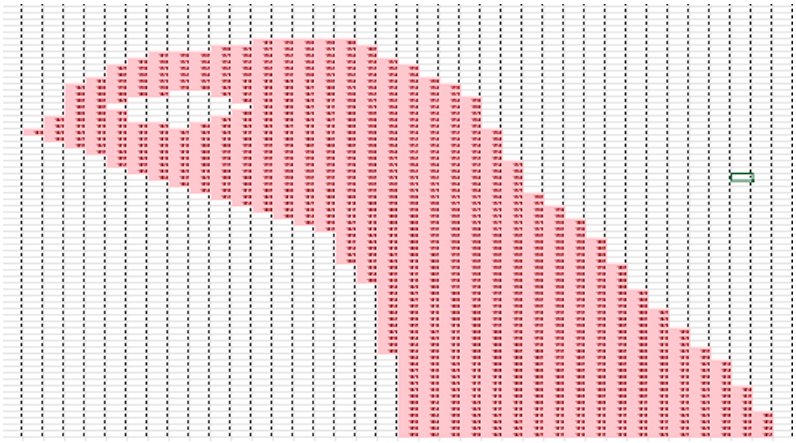
Figura 31 – Histograma da imagem recortada - Kinect 360 depth



Fonte: Autor

Observamos na Figura 31 que no primeiro máximo local, após o "0", tem 352 ocorrências na intensidade de pixel de 1400 a 1600 (Essas intensidades representam as distâncias do sensor até o corpo em "mm"). Este máximo local nos fornece uma posição, no caso 1400mm, este valor acrescido de 120mm será usado como ponto de corte para separarmos a região da mão do restante do corpo e do fundo. Fazendo os valores maiores que o ponto encontrado se tonarem "0", resulta a Figura 32. Os valores são exibidos numa planilha de excel para uma melhor visualização das distancias que foram explicados anteriormente.

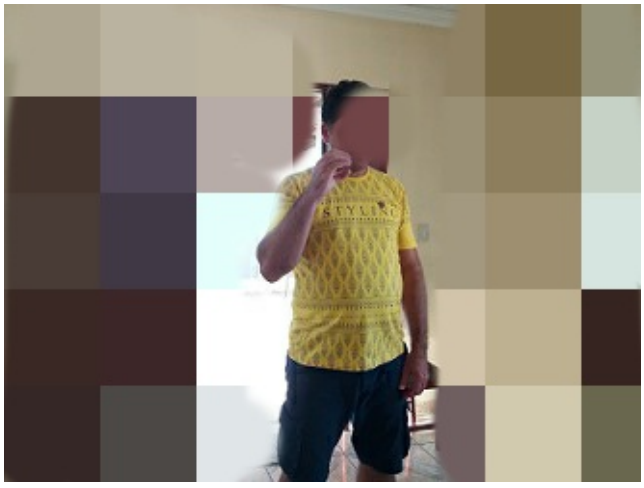
Figura 32 – Valores da imagem segmentada - Kinect 360 depth



Fonte: Autor.

No momento da aquisição um cuidado a ser tomado, é colocar o ator em um local onde o fundo da imagem esteja dentro do "*range*" do Kinect. Caso ocorra como na Figura 33 em que no fundo da imagem a porta estava aberta, o Kinect irá atribuir todos os pontos que passarem de seu limite iguais a "0".

Figura 33 – Imagem com fundo fora dos limites do Kinect



Fonte: Autor.

Para padronização das imagens capturadas será utilizada como base os gestos disponibilizados na página do INES.

Na aquisição para base de dados foram utilizados pessoas que não possuem domínio em LIBRAS para que o algoritmo tenha maior confiabilidade nos resultados.

Também foi utilizada a pesquisa documental e bibliográfica para informações sobre os métodos de aquisição, segmentação, segregação e extração de características de imagens.

A base de dados será montada através de imagens capturadas pelo sensor Kinect processadas e armazenadas pelo software Matlab.

4 RESULTADOS E DISCUSSÕES

Neste capitulo apresentamos os resultados obtidos.

4.1 RESULTADOS

Foram utilizadas a base de dados ASL *Finger Spelling Dataset*, disponível em: http://empslocal.ex.ac.uk/people/staff/np331/index.php?section=FingerSpellingDataset. Essa base contem todo o alfabeto, exceto os sinais J e Z. Estes não serão utilizados devido possuírem movimentos e o trabalho se propõem a sinais estáticos. Estes sinais são obtidos de cinco atores variando a posição da mão no momento da captura.

Da base de dados selecionamos primeiramente as vogais, sendo que utilizamos 2.495 imagens para o treino e 60 para o teste, ambas em RGB. Este é apresentado na tabela 5. Para cada ator temos seu desempenho para cada vogal.

Tabela 5 – Resultados com as vogais

	Letra(a)		**Letra(e)**		**Letra(i)**		**Letra(o)**		**Letra(u)**	
	Certo	Errado	C	E	C	E	C	E	C	E
Ator(A)	100%	0%	92%	8%	100%	0%	100%	0%	100%	0%
Ator(B)	75%	25%	92%	8%	83%	17%	83%	17%	92%	8%
Ator(C)	83%	17%	25%	75%	100%	0%	100%	0%	67%	33%
Ator(D)	75%	25%	25%	75%	75%	25%	92%	8%	75%	25%
Ator(E)	92%	8%	75%	25%	83%	17%	100%	0%	75%	25%

Fonte: Autor

A Tabela 6 apresenta a matrix de confusão. Utilizando o classificador ExemploBagOfWordsWebiniar, e o aplicativo Image Batch Processor no Matlab para testar o classificador. Onde obtemos uma acurácia de 88%. A vogal que apresentou melhor desempenho foi a letra "*U*" com acurácia de 92%.

Tabela 6 – Matrix de confusão das vogais

Letras	a	e	i	o	u
a	0,91	0,03	0,01	0,03	0,02
e	0,06	0,81	0,04	0,06	0,03
i	0,02	0,03	0,88	0,03	0,04
o	0,03	0,03	0,03	0,89	0,02
u	0,01	0,02	0,03	0,01	0,92

Acurácia = 0.88

Fonte: Autor

Com o objetivo de verificar o desempenho do classificador inserimos outras letras semelhantes, no caso as letra C e S. Com isto nossa matrix de confusão é apresentada na Tabela 7. Onde sua acurácia diminui para 82%. Verificando a linha da letra "*C*", podemos observar que a letra mais confundida foi a letra "*O*", tendo 5%, devido as mesmas serem parecidas. A letra "*U*" obteve novamente uma melhor acurácia, com o valor de 91%.

Tabela 7 – Matrix de confusão das letras a,c,e,i,o,s,u

Letras	a	c	e	i	o	s	u
a	0,83	0,06	0,03	0,02	0,02	0,04	0,01
c	0,03	0,84	0,03	0,00	0,05	0,03	0,01
e	0,06	0,05	0,72	0,04	0,04	0,05	0,03
i	0,01	0,01	0,03	0,87	0,03	0,02	0,04
o	0,04	0,04	0,04	0,03	0,79	0,04	0,02
s	0,06	0,02	0,05	0,03	0,07	0,77	0,02
u	0,00	0,01	0,02	0,03	0,01	0,02	0,91

Acurácia = 0.82

Fonte: Autor

Vamos fazer uma extratificação destes desempenhos por vogais para cada ator e cada imagem. Para uma melhor visualização no momento de testar o classificador inserimos duas letras no final de cada imagem. A última representa o ator (letra maiúscula) e a penúltima representa o letra que esta sendo testada (letra minuscula). Como podemos ver na Tabela 8 temos diferentes acurácias para cada ator. As imagens do ator A e B apresentaram somente uma imagem classificada errada, classificando como letra "e"e "i", respectivamente. Para as imagens do ator B tivemos três letras aparecendo na classificação, sendo a letra "e"a mais recorrente. Imagens do ator C, tiveram duas e ator letra D quatro.

Tabela 8 – Resultado de cada imagem para letra "a"de cada ator

Letra (a)									
Ator (A)		Ator (B)		Ator (C)		Ator (D)		Ator (E)	
IMAGEM (Input)	LETRA (Output)	IMAGEM (Input)	LETRA (Output)	IMAGEM (Input)	LETRA (Output)	IMAGEM (Input)	LETRA (Output)	IMAGEM (Input)	LETRA (Output)
color_0_0501_aA	a	color_0_0501_aB	e	color_0_0501_aC	i	color_0_0501_aD	c	color_0_0501_aE	a
color_0_0502_aA	a	color_0_0502_aB	e	color_0_0502_aC	a	color_0_0502_aD	a	color_0_0502_aE	a
color_0_0503_aA	a	color_0_0503_aB	e	color_0_0503_aC	s	color_0_0503_aD	e	color_0_0503_aE	a
color_0_0504_aA	a	color_0_0504_aB	i	color_0_0504_aC	s	color_0_0504_aD	e	color_0_0504_aE	i
color_0_0505_aA	a	color_0_0505_aB	e	color_0_0505_aC	s	color_0_0505_aD	o	color_0_0505_aE	a
color_0_0506_aA	e	color_0_0506_aB	i	color_0_0506_aC	a	color_0_0506_aD	a	color_0_0506_aE	a
color_0_0507_aA	a	color_0_0507_aB	s	color_0_0507_aC	i	color_0_0507_aD	e	color_0_0507_aE	a
color_0_0508_aA	a	color_0_0508_aB	e	color_0_0508_aC	a	color_0_0508_aD	e	color_0_0508_aE	a
color_0_0509_aA	a	color_0_0509_aB	a	color_0_0509_aC	a	color_0_0509_aD	a	color_0_0509_aE	a
color_0_0510_aA	a	color_0_0510_aB	a	color_0_0510_aC	a	color_0_0510_aD	o	color_0_0510_aE	a
color_0_0511_aA	a	color_0_0511_aB	a	color_0_0511_aC	a	color_0_0511_aD	o	color_0_0511_aE	a
color_0_0512_aA	a	color_0_0512_aB	a	color_0_0512_aC	a	color_0_0512_aD	c	color_0_0512_aE	a
Acurácia	92%	Acurácia	33%	Acurácia	58%	Acurácia	25%	Acurácia	92%

Fonte: Autor

Da Tabela 8 geramos a Tabela 9 onde representamos a porcentagem que cada letra é confundida pelo nosso classificador.

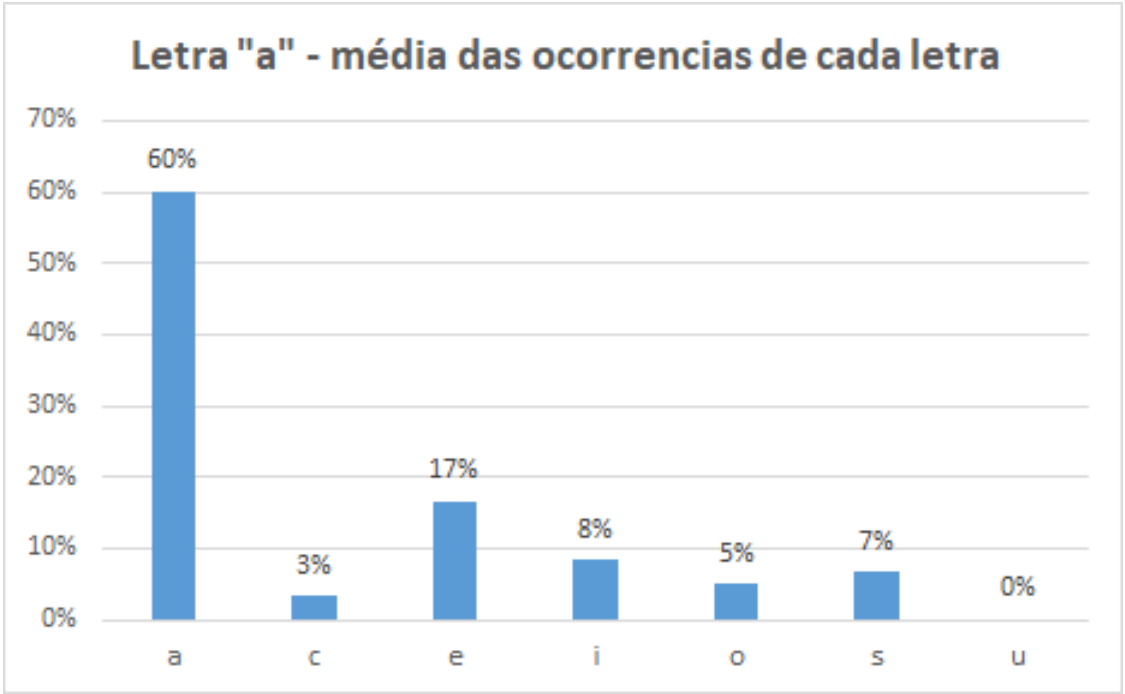
Tabela 9 – Resultado da letra "a"de cada ator

Letra (a)							
Atores	% de cada letra para cada ator						
	a	c	e	i	o	s	u
Ator (A)	92%	0%	8%	0%	0%	0%	0%
Ator (B)	33%	0%	42%	17%	0%	8%	0%
Ator (C)	58%	0%	0%	17%	0%	25%	0%
Ator (D)	25%	17%	33%	0%	25%	0%	0%
Ator (E)	92%	0%	0%	8%	0%	0%	0%
Media	60%	3%	17%	8%	5%	7%	0%

Fonte: Autor

Desta Tabela 9 geramos a Figura 34. Nesta imagem geramos graficamente a ocorrências de cada imagem, onde podemos verificar que a letra "*e*" é a mais confundida pelo classificador.

Figura 34 – Média das ocorrências de cada letra



Fonte: Autor

Na Tabela 10 temos apenas dois atores que suas imagens são confundidas com as demais. Para o ator D temos duas ocorrências para a letra "*o*" e para p ator E, temos duas letras, sendo que a letra "*e*" apresenta maior reincidência. O ator "*E*" apresenta menos acurácia no valor de 33%.

Tabela 10 – Resultado de cada imagem para letra "*c*" de cada ator

Letra (c)									
Ator (A)		Ator (B)		Ator (C)		Ator (D)		Ator (E)	
IMAGEM (Input)	LETRA (Output)	IMAGEM (Input)	LETRA (Output)	IMAGEM (Input)	LETRA (Output)	IMAGEM (Input)	LETRA (Output)	IMAGEM (Input)	LETRA (Output)
color_2_0501_cA	c	color_2_0501_cB	c	color_2_0501_cC	c	color_2_0501_cD	o	color_2_0501_cE	e
color_2_0502_cA	c	color_2_0502_cB	c	color_2_0502_cC	c	color_2_0502_cD	o	color_2_0502_cE	e
color_2_0503_cA	c	color_2_0503_cB	c	color_2_0503_cC	c	color_2_0503_cD	c	color_2_0503_cE	c
color_2_0504_cA	c	color_2_0504_cB	c	color_2_0504_cC	c	color_2_0504_cD	c	color_2_0504_cE	c
color_2_0505_cA	c	color_2_0505_cB	c	color_2_0505_cC	c	color_2_0505_cD	c	color_2_0505_cE	c
color_2_0506_cA	c	color_2_0506_cB	c	color_2_0506_cC	c	color_2_0506_cD	c	color_2_0506_cE	c
color_2_0507_cA	c	color_2_0507_cB	c	color_2_0507_cC	c	color_2_0507_cD	c	color_2_0507_cE	e
color_2_0508_cA	c	color_2_0508_cB	c	color_2_0508_cC	c	color_2_0508_cD	c	color_2_0508_cE	i
color_2_0509_cA	c	color_2_0509_cB	c	color_2_0509_cC	c	color_2_0509_cD	c	color_2_0509_cE	i
color_2_0510_cA	c	color_2_0510_cB	c	color_2_0510_cC	c	color_2_0510_cD	c	color_2_0510_cE	e
color_2_0511_cA	c	color_2_0511_cB	c	color_2_0511_cC	c	color_2_0511_cD	c	color_2_0511_cE	e
color_2_0512_cA	c	color_2_0512_cB	c	color_2_0512_cC	c	color_2_0512_cD	c	color_2_0512_cE	e
Acurácia	100%	Acurácia	100%	Acurácia	100%	Acurácia	83%	Acurácia	33%

Fonte: Autor

Na Tabela 11 temos apenas o ator A com acurácia de 100%. Para o ator B aparece somente duas ocorrências erradas, sendo uma de cada letra. No ator E aparece também somente duas letras, sendo que a letra "*o*" tem três ocorrências. O ator D tem três ocorrências e no ator B como o menor desempenho aparece duas letras, sendo a letra "*u*" com cinco ocorrências.

Tabela 11 – Resultado de cada imagem para letra "*e*" de cada ator

Letra (e)									
Ator (A)		Ator (B)		Ator (C)		Ator (D)		Ator (E)	
IMAGEM (Input)	LETRA (Output)	IMAGEM (Input)	LETRA (Output)	IMAGEM (Input)	LETRA (Output)	IMAGEM (Input)	LETRA (Output)	IMAGEM (Input)	LETRA (Output)
color_4_0501_eA	e	color_4_0501_eB	e	color_4_0501_eC	u	color_4_0501_eD	e	color_4_0501_eE	u
color_4_0502_eA	e	color_4_0502_eB	e	color_4_0502_eC	u	color_4_0502_eD	u	color_4_0502_eE	o
color_4_0503_eA	e	color_4_0503_eB	e	color_4_0503_eC	e	color_4_0503_eD	o	color_4_0503_eE	e
color_4_0504_eA	e	color_4_0504_eB	o	color_4_0504_eC	u	color_4_0504_eD	i	color_4_0504_eE	o
color_4_0505_eA	e	color_4_0505_eB	e	color_4_0505_eC	i	color_4_0505_eD	o	color_4_0505_eE	o
color_4_0506_eA	e	color_4_0506_eB	c	color_4_0506_eC	u	color_4_0506_eD	e	color_4_0506_eE	e
color_4_0507_eA	e	color_4_0507_eB	e	color_4_0507_eC	i	color_4_0507_eD	u	color_4_0507_eE	e
color_4_0508_eA	e	color_4_0508_eB	e	color_4_0508_eC	u	color_4_0508_eD	e	color_4_0508_eE	e
color_4_0509_eA	e	color_4_0509_eB	e	color_4_0509_eC	i	color_4_0509_eD	i	color_4_0509_eE	e
color_4_0510_eA	e	color_4_0510_eB	e	color_4_0510_eC	i	color_4_0510_eD	e	color_4_0510_eE	e
color_4_0511_eA	e	color_4_0511_eB	e	color_4_0511_eC	e	color_4_0511_eD	e	color_4_0511_eE	e
color_4_0512_eA	e	color_4_0512_eB	e	color_4_0512_eC	e	color_4_0512_eD	e	color_4_0512_eE	e
Acurácia	100%	Acurácia	83%	Acurácia	25%	Acurácia	50%	Acurácia	67%

Fonte: Autor

Na Tabela 12 temos os atores A e C com acurácia de 100%. Para o ator E com acurácia de 83% uma letra com duas ocorrências. O ator D com uma letra e três ocorrências e o ator B com menor desempenho com três letras.

Tabela 12 – Resultado de cada imagem para letra "i"de cada ator

Letra (i)									
Ator (A)		Ator (B)		Ator (C)		Ator (D)		Ator (E)	
IMAGEM (Input)	LETRA (Output)	IMAGEM (Input)	LETRA (Output)	IMAGEM (Input)	LETRA (Output)	IMAGEM (Input)	LETRA (Output)	IMAGEM (Input)	LETRA (Output)
color_8_0501_iA	i	color_8_0501_iB	s	color_8_0501_iC	i	color_8_0501_iD	e	color_8_0501_iE	i
color_8_0502_iA	i	color_8_0502_iB	o	color_8_0502_iC	i	color_8_0502_iD	e	color_8_0502_iE	i
color_8_0503_iA	i	color_8_0503_iB	i	color_8_0503_iC	i	color_8_0503_iD	e	color_8_0503_iE	i
color_8_0504_iA	i	color_8_0504_iB	i	color_8_0504_iC	i	color_8_0504_iD	i	color_8_0504_iE	i
color_8_0505_iA	i	color_8_0505_iB	u	color_8_0505_iC	i	color_8_0505_iD	i	color_8_0505_iE	i
color_8_0506_iA	i	color_8_0506_iB	u	color_8_0506_iC	i	color_8_0506_iD	i	color_8_0506_iE	i
color_8_0507_iA	i	color_8_0507_iB	s	color_8_0507_iC	i	color_8_0507_iD	i	color_8_0507_iE	i
color_8_0508_iA	i	color_8_0508_iB	i	color_8_0508_iC	i	color_8_0508_iD	i	color_8_0508_iE	i
color_8_0509_iA	i	color_8_0509_iB	i	color_8_0509_iC	i	color_8_0509_iD	i	color_8_0509_iE	o
color_8_0510_iA	i	color_8_0510_iB	i	color_8_0510_iC	i	color_8_0510_iD	i	color_8_0510_iE	o
color_8_0511_iA	i	color_8_0511_iB	i	color_8_0511_iC	i	color_8_0511_iD	i	color_8_0511_iE	i
color_8_0512_iA	i	color_8_0512_iB	i	color_8_0512_iC	i	color_8_0512_iD	i	color_8_0512_iE	i
Acurácia	100%	Acurácia	58%	Acurácia	100%	Acurácia	75%	Acurácia	83%

Fonte: Autor

Na Tabela 13 temos os atores A e C com acurácias de 100%. Para o ator E temos somente uma letra aparecendo com uma ocorrência. O ator D com duas ocorrências de uma letra e o ator B com três letras.

Tabela 13 – Resultado de cada imagem para letra "o"de cada ator

Letra (o)									
Ator (A)		Ator (B)		Ator (C)		Ator (D)		Ator (E)	
IMAGEM (Input)	LETRA (Output)	IMAGEM (Input)	LETRA (Output)	IMAGEM (Input)	LETRA (Output)	IMAGEM (Input)	LETRA (Output)	IMAGEM (Input)	LETRA (Output)
color_14_0501_oA	o	color_14_0501_oB	o	color_14_0501_oC	o	color_14_0501_oD	o	color_14_0501_oE	o
color_14_0502_oA	o	color_14_0502_oB	i	color_14_0502_oC	o	color_14_0502_oD	s	color_14_0502_oE	c
color_14_0503_oA	o	color_14_0503_oB	o	color_14_0503_oC	o	color_14_0503_oD	i	color_14_0503_oE	o
color_14_0504_oA	o	color_14_0504_oB	c	color_14_0504_oC	o	color_14_0504_oD	i	color_14_0504_oE	o
color_14_0505_oA	o	color_14_0505_oB	a	color_14_0505_oC	o	color_14_0505_oD	o	color_14_0505_oE	o
color_14_0506_oA	o	color_14_0506_oB	o	color_14_0506_oC	o	color_14_0506_oD	o	color_14_0506_oE	o
color_14_0507_oA	o	color_14_0507_oB	o	color_14_0507_oC	o	color_14_0507_oD	o	color_14_0507_oE	o
color_14_0508_oA	o	color_14_0508_oB	o	color_14_0508_oC	o	color_14_0508_oD	o	color_14_0508_oE	o
color_14_0509_oA	o	color_14_0509_oB	c	color_14_0509_oC	o	color_14_0509_oD	o	color_14_0509_oE	o
color_14_0510_oA	o	color_14_0510_oB	c	color_14_0510_oC	o	color_14_0510_oD	o	color_14_0510_oE	o
color_14_0511_oA	o	color_14_0511_oB	o	color_14_0511_oC	o	color_14_0511_oD	o	color_14_0511_oE	s
color_14_0512_oA	o	color_14_0512_oB	o	color_14_0512_oC	o	color_14_0512_oD	o	color_14_0512_oE	c
Acurácia	100%	Acurácia	58%	Acurácia	100%	Acurácia	75%	Acurácia	75%

Fonte: Autor

Na Tabela 14 temos os atores B e C com acurácias de 100%, sendo que no ator C so foram utilizadas sete imagens para o teste. Para o ator E temos duas letras, sendo que a letra "e"tem cinco ocorrências. Para o ator D, temos duas letras, sendo a letra "o"com seis ocorrências. Para o ator "a"com menor acurácia, temos quatro letras, sendo que a letra "o"tem oito ocorrências.

Tabela 14 – Resultado de cada imagem para letra "s"de cada ator

Letra (s)									
Ator (A)		Ator (B)		Ator (C)		Ator (D)		Ator (E)	
IMAGEM (Input)	LETRA (Output)	IMAGEM (Input)	LETRA (Output)	IMAGEM (Input)	LETRA (Output)	IMAGEM (Input)	LETRA (Output)	IMAGEM (Input)	LETRA (Output)
color_18_0501_sA	o	color_18_0501_sB	s	color_18_0501_sC	s	color_18_0501_sD	u	color_18_0501_sE	e
color_18_0502_sA	o	color_18_0502_sB	s	color_18_0502_sC	s	color_18_0502_sD	u	color_18_0502_sE	u
color_18_0503_sA	o	color_18_0503_sB	s	color_18_0503_sC	s	color_18_0503_sD	s	color_18_0503_sE	s
color_18_0504_sA	o	color_18_0504_sB	s	color_18_0504_sC	s	color_18_0504_sD	s	color_18_0504_sE	s
color_18_0505_sA	o	color_18_0505_sB	s	color_18_0505_sC	s	color_18_0505_sD	u	color_18_0505_sE	s
color_18_0506_sA	o	color_18_0506_sB	s	color_18_0506_sC	s	color_18_0506_sD	s	color_18_0506_sE	a
color_18_0507_sA	o	color_18_0507_sB	s	color_18_0507_sC	s	color_18_0507_sD	o	color_18_0507_sE	e
color_18_0508_sA	s	color_18_0508_sB	s			color_18_0508_sD	o	color_18_0508_sE	e
color_18_0509_sA	i	color_18_0509_sB	s			color_18_0509_sD	o	color_18_0509_sE	e
color_18_0510_sA	e	color_18_0510_sB	s			color_18_0510_sD	o	color_18_0510_sE	e
color_18_0511_sA	a	color_18_0511_sB	s			color_18_0511_sD	o	color_18_0511_sE	s
color_18_0512_sA	s	color_18_0512_sB	s			color_18_0512_sD	o	color_18_0512_sE	s
Acurácia	17%	Acurácia	100%	Acurácia	100%	Acurácia	25%	Acurácia	42%

Fonte: Autor.

Na Tabela 15 temos os atores B e C com acurácia de 100%. Para o ator A, temos uma letra com apenas uma ocorrência. Para os atores D e E com uma letra e duas ocorrências.

Tabela 15 – Resultado de cada imagem para letra "u"de cada ator

Letra (u)									
Ator (A)		Ator (B)		Ator (C)		Ator (D)		Ator (E)	
IMAGEM (Input)	LETRA (Output)	IMAGEM (Input)	LETRA (Output)	IMAGEM (Input)	LETRA (Output)	IMAGEM (Input)	LETRA (Output)	IMAGEM (Input)	LETRA (Output)
color_20_0501_uA	u	color_20_0501_uB	u	color_20_0501_uC	u	color_20_0501_uD	u	color_20_0501_uE	c
color_20_0502_uA	u	color_20_0502_uB	u	color_20_0502_uC	u	color_20_0502_uD	u	color_20_0502_uE	c
color_20_0503_uA	u	color_20_0503_uB	u	color_20_0503_uC	u	color_20_0503_uD	u	color_20_0503_uE	u
color_20_0504_uA	i	color_20_0504_uB	u	color_20_0504_uC	u	color_20_0504_uD	u	color_20_0504_uE	u
color_20_0505_uA	u	color_20_0505_uB	u	color_20_0505_uC	u	color_20_0505_uD	u	color_20_0505_uE	u
color_20_0506_uA	u	color_20_0506_uB	u	color_20_0506_uC	u	color_20_0506_uD	u	color_20_0506_uE	u
color_20_0507_uA	u	color_20_0507_uB	u	color_20_0507_uC	u	color_20_0507_uD	i	color_20_0507_uE	u
color_20_0508_uA	u	color_20_0508_uB	u	color_20_0508_uC	u	color_20_0508_uD	u	color_20_0508_uE	u
color_20_0509_uA	u	color_20_0509_uB	u	color_20_0509_uC	u	color_20_0509_uD	u	color_20_0509_uE	u
color_20_0510_uA	u	color_20_0510_uB	u	color_20_0510_uC	u	color_20_0510_uD	u	color_20_0510_uE	u
color_20_0511_uA	u	color_20_0511_uB	u	color_20_0511_uC	u	color_20_0511_uD	i	color_20_0511_uE	u
color_20_0512_uA	u	color_20_0512_uB	u	color_20_0512_uC	u	color_20_0512_uD	u	color_20_0512_uE	u
Acurácia	92%	Acurácia	100%	Acurácia	100%	Acurácia	83%	Acurácia	83%

Fonte: Autor.

Na tabela 16 apresenta a matrix de confusão de quase todas as letras. Como pode ser visto houve uma diminuição na acurácia de 88% somente com as vogais para 59% utilizando a maioria das letras.

Tabela 16 – Matrix de confusão das letras

Letras	a	b	c	d	e	f	g	i	l	m	n	o	p	q	r	s	t	u	v
a	0,64	0,02	0,10	0,01	0,01	0,00	0,03	0,01	0,01	0,03	0,05	0,01	0,02	0,01	0,00	0,02	0,02	0,00	0,00
b	0,00	0,75	0,02	0,02	0,01	0,03	0,01	0,01	0,00	0,01	0,04	0,01	0,00	0,01	0,02	0,01	0,00	0,03	0,02
c	0,02	0,02	0,78	0,01	0,01	0,01	0,01	0,00	0,00	0,01	0,05	0,03	0,01	0,02	0,01	0,01	0,01	0,00	0,00
d	0,00	0,02	0,02	0,60	0,02	0,04	0,00	0,03	0,03	0,01	0,06	0,02	0,01	0,00	0,03	0,01	0,02	0,03	0,06
e	0,08	0,04	0,06	0,05	0,32	0,01	0,01	0,02	0,03	0,05	0,09	0,09	0,01	0,01	0,03	0,07	0,02	0,02	0,01
f	0,00	0,03	0,01	0,02	0,00	0,78	0,00	0,02	0,01	0,00	0,04	0,00	0,01	0,00	0,03	0,00	0,00	0,01	0,03
g	0,01	0,00	0,00	0,00	0,00	0,00	0,92	0,00	0,02	0,00	0,01	0,00	0,00	0,00	0,00	0,00	0,01	0,00	0,00
i	0,00	0,02	0,01	0,04	0,01	0,04	0,01	0,55	0,07	0,02	0,08	0,02	0,02	0,02	0,03	0,01	0,01	0,03	0,02
l	0,00	0,00	0,00	0,01	0,01	0,00	0,03	0,02	0,84	0,01	0,04	0,00	0,01	0,00	0,01	0,00	0,00	0,00	0,01
m	0,02	0,01	0,01	0,01	0,03	0,01	0,00	0,02	0,00	0,62	0,09	0,06	0,02	0,03	0,01	0,04	0,01	0,01	0,01
n	0,04	0,02	0,03	0,02	0,04	0,01	0,01	0,02	0,02	0,18	0,32	0,05	0,04	0,03	0,02	0,03	0,08	0,03	0,01
o	0,01	0,01	0,05	0,01	0,02	0,01	0,01	0,01	0,00	0,05	0,08	0,62	0,02	0,02	0,01	0,05	0,01	0,01	0,00
p	0,01	0,02	0,01	0,00	0,01	0,01	0,02	0,01	0,01	0,02	0,06	0,02	0,68	0,06	0,01	0,01	0,02	0,01	0,00
q	0,02	0,02	0,03	0,01	0,02	0,02	0,02	0,02	0,01	0,02	0,08	0,04	0,10	0,50	0,01	0,02	0,02	0,01	0,02
r	0,01	0,04	0,01	0,11	0,02	0,10	0,01	0,03	0,03	0,01	0,07	0,00	0,02	0,03	0,20	0,01	0,02	0,14	0,16
s	0,04	0,01	0,02	0,01	0,01	0,00	0,01	0,02	0,00	0,06	0,07	0,06	0,02	0,01	0,01	0,63	0,02	0,01	0,00
t	0,13	0,03	0,02	0,01	0,02	0,00	0,03	0,02	0,03	0,04	0,11	0,01	0,11	0,03	0,02	0,08	0,28	0,01	0,01
u	0,00	0,04	0,02	0,04	0,01	0,02	0,00	0,01	0,00	0,01	0,05	0,01	0,01	0,00	0,04	0,02	0,00	0,62	0,09
v	0,00	0,02	0,01	0,03	0,01	0,05	0,00	0,03	0,01	0,01	0,04	0,00	0,01	0,00	0,04	0,01	0,00	0,07	0,64

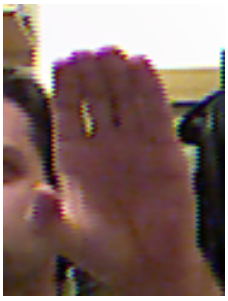
Acurácia = 0.59

Fonte: Autor.

4.2 Teste com imagens selecionadas

No banco de dados encontramos imagens que inicialmente não representam a diretamente a letra, no caso de a mão estar numa posição bem diferente do que simboliza a letra. No caso da Fig. 35 que a mão esta bem rotacionada. Deste modo retiramos estas imagens da nossa banco de trabalho.

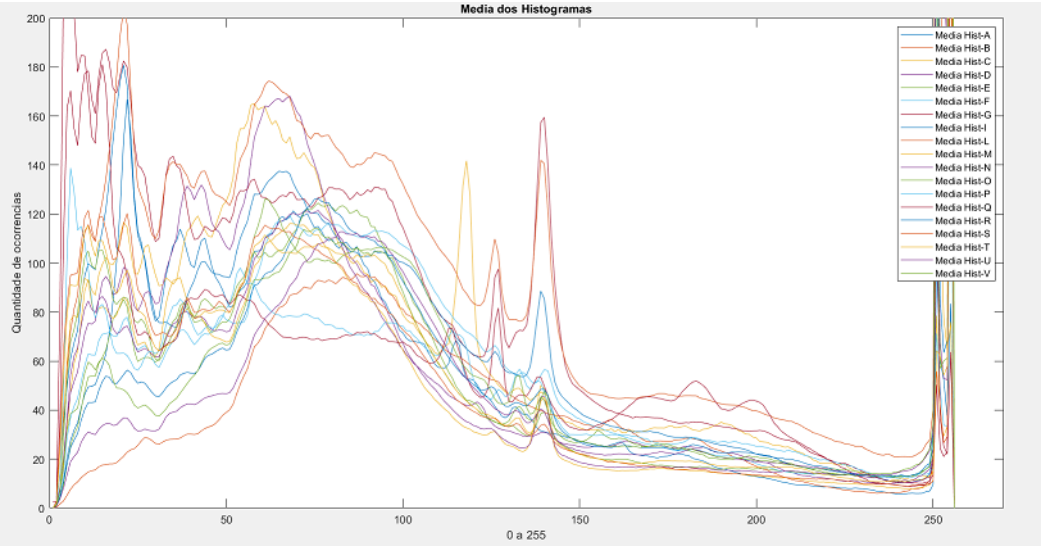
Figura 35 – Imagem da letra "C"do ator "A"



Fonte: ASL Finger Spelling Dataset.

A Figura 36 exibe as curvas da media dos histogramas de cada letra. Podemos observar que algumas curvas particulares apresentam uma distorção em alguns pontos do gráfico. Esses pontos nos dispertam curiosidade e vamos analisa-los mais a frente.

Figura 36 – Media dos histogramas selecionadas



Fonte: Autor.

Apos as seleção das imagens obtemos a Tabela 17. Houve um aumento na acurácia de 59% para 64%. Com o objetivo de melhorarmos nosso classificador iremos estudar alguns casos em particular, fazer algumas mudanças e avaliar os resultados.

Tabela 17 – Matrix de confusão de todas as letras selecionadas

Letras	a	b	c	d	e	f	g	i	l	m	n	o	p	q	r	s	t	u	v
a	0,71	0,00	0,06	0,00	0,00	0,01	0,02	0,00	0,02	0,03	0,06	0,02	0,01	0,02	0,00	0,02	0,03	0,00	0,00
b	0,00	0,82	0,01	0,01	0,00	0,03	0,00	0,01	0,01	0,01	0,04	0,01	0,01	0,00	0,01	0,00	0,00	0,02	0,01
c	0,02	0,00	0,77	0,00	0,01	0,00	0,03	0,00	0,03	0,02	0,01	0,06	0,01	0,01	0,00	0,02	0,01	0,00	0,00
d	0,00	0,01	0,02	0,74	0,01	0,02	0,00	0,01	0,03	0,00	0,05	0,04	0,01	0,00	0,03	0,00	0,00	0,02	0,02
e	0,06	0,01	0,06	0,04	0,50	0,00	0,01	0,02	0,04	0,05	0,06	0,04	0,01	0,00	0,01	0,06	0,02	0,01	0,00
f	0,00	0,03	0,00	0,02	0,00	0,79	0,00	0,03	0,03	0,00	0,03	0,00	0,01	0,00	0,01	0,00	0,00	0,01	0,04
g	0,01	0,00	0,01	0,00	0,00	0,00	0,94	0,00	0,02	0,00	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
i	0,00	0,02	0,02	0,02	0,03	0,04	0,01	0,50	0,08	0,01	0,10	0,02	0,04	0,01	0,03	0,02	0,00	0,02	0,03
l	0,00	0,01	0,02	0,01	0,01	0,01	0,05	0,00	0,83	0,00	0,04	0,00	0,00	0,00	0,00	0,01	0,01	0,00	0,01
m	0,03	0,01	0,00	0,02	0,03	0,00	0,00	0,02	0,01	0,68	0,09	0,05	0,01	0,02	0,01	0,03	0,01	0,01	0,00
n	0,03	0,02	0,03	0,04	0,06	0,04	0,01	0,02	0,03	0,18	0,29	0,06	0,03	0,01	0,02	0,03	0,09	0,02	0,00
o	0,02	0,01	0,06	0,01	0,04	0,01	0,02	0,01	0,00	0,04	0,06	0,65	0,00	0,01	0,00	0,06	0,00	0,00	0,00
p	0,01	0,01	0,01	0,02	0,00	0,02	0,01	0,00	0,01	0,01	0,06	0,01	0,66	0,09	0,01	0,02	0,02	0,01	0,00
q	0,00	0,01	0,03	0,01	0,01	0,01	0,01	0,02	0,01	0,00	0,02	0,00	0,04	0,79	0,00	0,01	0,01	0,01	0,01
r	0,00	0,06	0,00	0,08	0,02	0,09	0,00	0,04	0,04	0,01	0,07	0,03	0,03	0,02	0,21	0,01	0,01	0,15	0,14
s	0,03	0,01	0,05	0,01	0,02	0,00	0,02	0,00	0,00	0,04	0,05	0,02	0,01	0,01	0,00	0,70	0,01	0,01	0,00
t	0,08	0,04	0,02	0,03	0,02	0,02	0,04	0,01	0,04	0,09	0,09	0,04	0,04	0,02	0,01	0,05	0,36	0,00	0,00
u	0,00	0,08	0,01	0,05	0,01	0,03	0,00	0,02	0,00	0,01	0,03	0,01	0,01	0,01	0,04	0,02	0,00	0,61	0,07
v	0,00	0,03	0,01	0,03	0,01	0,05	0,00	0,01	0,01	0,01	0,03	0,01	0,03	0,01	0,02	0,00	0,00	0,10	0,64

Acurácia = 0,64

Fonte: Autor.

4.3 Teste com imagens selecionadas pelo histograma

Como pode ser visto na Tabela 17 as letra r,u e v são confundidas pelo classificador. Observando a Fig. 37 vemos que elas são semelhantes. Para melhorar nosso classificador vamos fazer uma verificação nas mesmas.

Figura 37 – Letras R, U e V

Fonte: (FUNCAP, 2015)

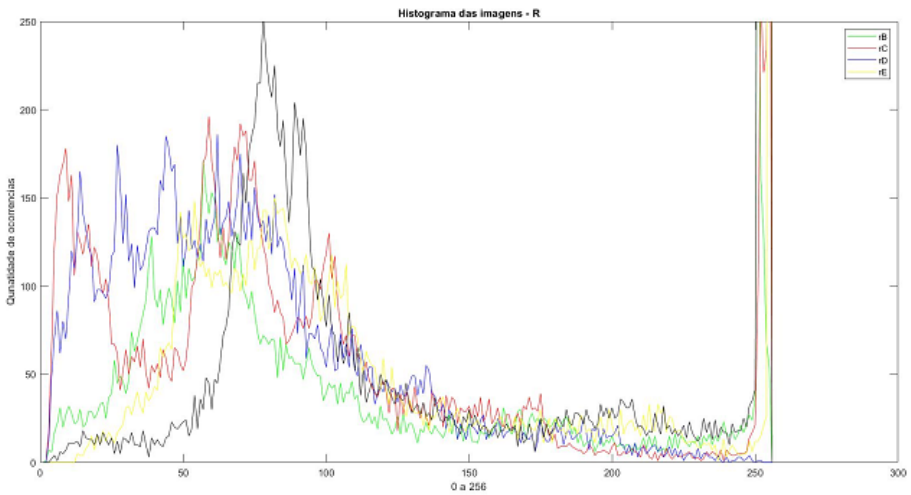
O banco de dados contem varias imagens dos cinco atores. Iremos pegar cinco imagens, uma para cada ator, em um intervalo de vinte e cinco imagens extrair o histograma e exibi-los de forma gráfica. Para exibi-las somente as curvas, vamos converter as imagens para escala de cinza. As imagens são no formato "png", classe "uint8".

4.3.1 Seleção das imagens baseada nos histogramas divergentes

Nesta parte do trabalho extraímos os histogramas de várias imagens da Letra "R"e colocamos de forma gráfica para uma melhor visualização para tomada de decisão.

Na Fig. 38 podemos observar que as imagens da letra R dos vários atores não tem curvas similares, são bem divergentes, dificultando o trabalho do nosso classificador.

Figura 38 – Histograma da letra "R"(conjunto 350)



Fonte: Autor

Após excluir as imagens bem distintas obtemos a matriz de confusão da Tabela 18. Verificamos que houve uma aumento na acurácia de letra "R" para 24%, enquanto que a letra "U" foi mais confundida subindo sua acurácia para 16% e a letra "V" foi menos confundida e obteve sua acurácia de 11%.

Tabela 18 – Matrix de confusão das imagens selecionadas e histograma confuso

Letras	a	b	c	d	e	f	g	i	l	m	n	o	p	q	r	s	t	u	v
a	0,68	0,01	0,09	0,03	0,02	0,00	0,01	0,00	0,00	0,03	0,05	0,01	0,01	0,01	0,01	0,02	0,01	0,01	0,00
b	0,01	0,82	0,00	0,01	0,01	0,02	0,00	0,00	0,01	0,01	0,02	0,02	0,01	0,00	0,01	0,01	0,00	0,02	0,01
c	0,02	0,00	0,77	0,00	0,02	0,00	0,03	0,00	0,02	0,00	0,07	0,02	0,00	0,02	0,00	0,01	0,01	0,00	0,00
d	0,01	0,00	0,02	0,75	0,01	0,03	0,00	0,01	0,01	0,01	0,06	0,01	0,01	0,00	0,01	0,01	0,02	0,01	0,03
e	0,02	0,03	0,04	0,02	0,58	0,00	0,00	0,01	0,03	0,03	0,10	0,04	0,02	0,00	0,02	0,02	0,01	0,02	0,01
f	0,00	0,04	0,00	0,03	0,00	0,82	0,00	0,02	0,01	0,00	0,03	0,00	0,01	0,01	0,01	0,00	0,00	0,01	0,03
g	0,01	0,00	0,00	0,00	0,01	0,01	0,92	0,00	0,02	0,00	0,02	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,00
i	0,00	0,02	0,01	0,02	0,02	0,05	0,01	0,54	0,08	0,03	0,08	0,01	0,01	0,01	0,02	0,04	0,01	0,02	0,03
l	0,00	0,00	0,02	0,01	0,00	0,01	0,03	0,01	0,85	0,00	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,02	0,01
m	0,03	0,01	0,01	0,02	0,02	0,00	0,03	0,02	0,00	0,64	0,07	0,05	0,02	0,01	0,01	0,03	0,02	0,01	0,01
n	0,07	0,04	0,02	0,04	0,06	0,02	0,02	0,02	0,03	0,15	0,27	0,05	0,04	0,03	0,01	0,04	0,08	0,01	0,02
o	0,03	0,01	0,03	0,02	0,03	0,01	0,03	0,00	0,01	0,03	0,03	0,72	0,00	0,01	0,00	0,03	0,01	0,00	0,01
p	0,00	0,01	0,01	0,01	0,00	0,00	0,00	0,01	0,02	0,03	0,07	0,00	0,73	0,07	0,02	0,00	0,00	0,01	0,00
q	0,01	0,01	0,03	0,01	0,01	0,02	0,02	0,00	0,01	0,01	0,03	0,01	0,05	0,78	0,00	0,01	0,01	0,00	0,00
r	0,01	0,05	0,01	0,10	0,01	0,09	0,00	0,05	0,03	0,01	0,06	0,02	0,03	0,00	0,24	0,00	0,01	0,16	0,11
s	0,06	0,02	0,04	0,01	0,04	0,01	0,01	0,02	0,01	0,06	0,06	0,08	0,00	0,03	0,00	0,55	0,00	0,01	0,00
t	0,13	0,03	0,00	0,02	0,01	0,01	0,03	0,01	0,03	0,07	0,11	0,04	0,06	0,05	0,01	0,05	0,31	0,01	0,02
u	0,01	0,02	0,01	0,04	0,02	0,02	0,01	0,02	0,01	0,00	0,03	0,03	0,02	0,00	0,04	0,00	0,01	0,65	0,07
v	0,00	0,03	0,00	0,01	0,01	0,06	0,00	0,02	0,02	0,00	0,04	0,00	0,03	0,01	0,04	0,00	0,01	0,07	0,65

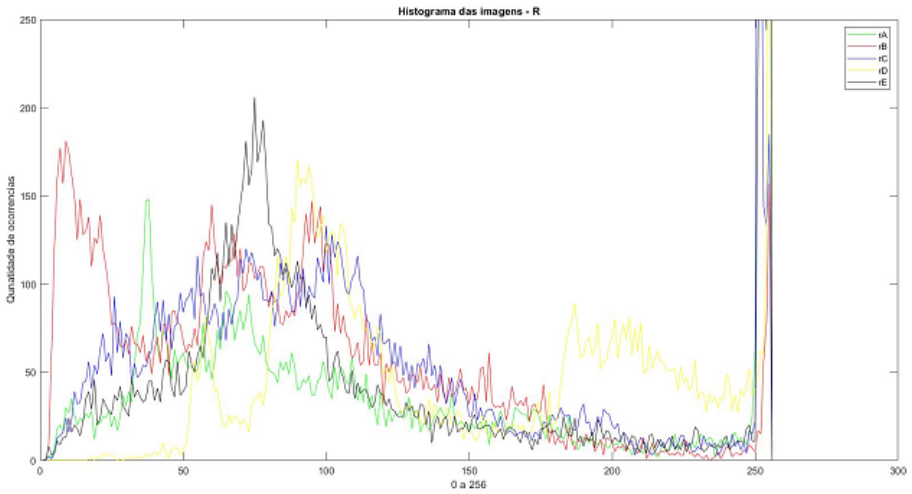
Acurácia = 0.64

Fonte: Autor

4.3.2 Seleção das imagens baseada no histograma semelhante

Na Fig. 39 a curva em vermelho (letra r do ator B) e a curva em amarelo (letra r do ator D) estão se distanciando no inicio das demais.

Figura 39 – Histograma da letra "R"(conjunto 125)



Fonte: Autor

Apos excluir as imagens que estavam se distanciando das demais curvas obtemos a matrix de confusão que pode ser visto na Tabela 19. Verificamos que houve uma diminuição na acurácia de letra "R" para 20%, a letra "U" obteve a acurácia de 17% e a letra "V" uma acurácia de 15%.

Tabela 19 – Matrix de confusão das imagens selecionadas e histograma semelhante

Letras	a	b	c	d	e	f	g	i	l	m	n	o	p	q	r	s	t	u	v
a	0,65	0,01	0,08	0,02	0,01	0,00	0,02	0,00	0,01	0,02	0,04	0,03	0,01	0,01	0,03	0,05	0,02	0,01	0,00
b	0,01	0,83	0,00	0,01	0,00	0,02	0,00	0,01	0,02	0,00	0,02	0,01	0,01	0,01	0,01	0,00	0,00	0,03	0,01
c	0,01	0,01	0,79	0,00	0,01	0,00	0,02	0,00	0,01	0,01	0,05	0,03	0,00	0,01	0,01	0,01	0,00	0,00	0,00
d	0,00	0,01	0,02	0,71	0,01	0,01	0,01	0,00	0,02	0,02	0,05	0,02	0,02	0,00	0,03	0,01	0,01	0,03	0,03
e	0,07	0,04	0,06	0,04	0,34	0,01	0,01	0,00	0,03	0,08	0,06	0,05	0,01	0,01	0,03	0,10	0,03	0,02	0,01
f	0,00	0,04	0,00	0,01	0,00	0,83	0,00	0,01	0,01	0,00	0,02	0,01	0,01	0,00	0,02	0,00	0,00	0,01	0,03
g	0,01	0,00	0,00	0,00	0,01	0,01	0,93	0,00	0,00	0,00	0,02	0,00	0,01	0,01	0,00	0,00	0,00	0,00	0,00
i	0,01	0,04	0,00	0,04	0,01	0,09	0,02	0,40	0,12	0,01	0,07	0,00	0,04	0,01	0,04	0,03	0,01	0,01	0,03
l	0,00	0,01	0,01	0,00	0,00	0,02	0,02	0,02	0,88	0,00	0,00	0,00	0,00	0,00	0,01	0,00	0,00	0,00	0,01
m	0,02	0,00	0,01	0,00	0,01	0,00	0,00	0,02	0,00	0,69	0,09	0,01	0,01	0,04	0,02	0,04	0,02	0,01	0,00
n	0,04	0,02	0,02	0,04	0,04	0,02	0,01	0,01	0,02	0,21	0,29	0,03	0,05	0,02	0,03	0,05	0,07	0,02	0,03
o	0,03	0,01	0,09	0,02	0,01	0,03	0,04	0,00	0,00	0,07	0,05	0,55	0,00	0,01	0,01	0,06	0,01	0,00	0,00
p	0,02	0,01	0,00	0,00	0,01	0,01	0,01	0,02	0,00	0,00	0,07	0,00	0,72	0,07	0,02	0,00	0,01	0,02	0,00
q	0,01	0,01	0,01	0,01	0,00	0,01	0,02	0,01	0,02	0,01	0,04	0,01	0,06	0,76	0,01	0,01	0,00	0,00	0,01
r	0,01	0,05	0,01	0,10	0,01	0,09	0,00	0,01	0,04	0,01	0,05	0,01	0,02	0,03	0,20	0,01	0,01	0,17	0,15
s	0,04	0,02	0,01	0,01	0,01	0,00	0,00	0,00	0,00	0,03	0,06	0,03	0,01	0,01	0,00	0,74	0,02	0,01	0,00
t	0,09	0,04	0,02	0,03	0,02	0,01	0,03	0,01	0,02	0,07	0,07	0,04	0,06	0,05	0,03	0,08	0,34	0,01	0,00
u	0,00	0,05	0,02	0,06	0,00	0,04	0,00	0,00	0,00	0,01	0,05	0,00	0,02	0,00	0,05	0,01	0,00	0,61	0,07
v	0,00	0,02	0,01	0,03	0,00	0,05	0,00	0,01	0,01	0,00	0,03	0,01	0,01	0,01	0,05	0,01	0,00	0,07	0,68

Acurácia = 0.63

Fonte: Autor

Os resultados para as letras "R", "U" e "V" são exibidos na Tabela 20, sendo o melhor resultado apresentado pela letra "R", que saiu de uma acuracidade de 20% com o conjunto de imagens que foram selecionadas pela rotação em que a mão se encontrava e foi para 28% com a eliminação de imagens com histogramas semelhantes.

Tabela 20 – Resultados após seleção histograma

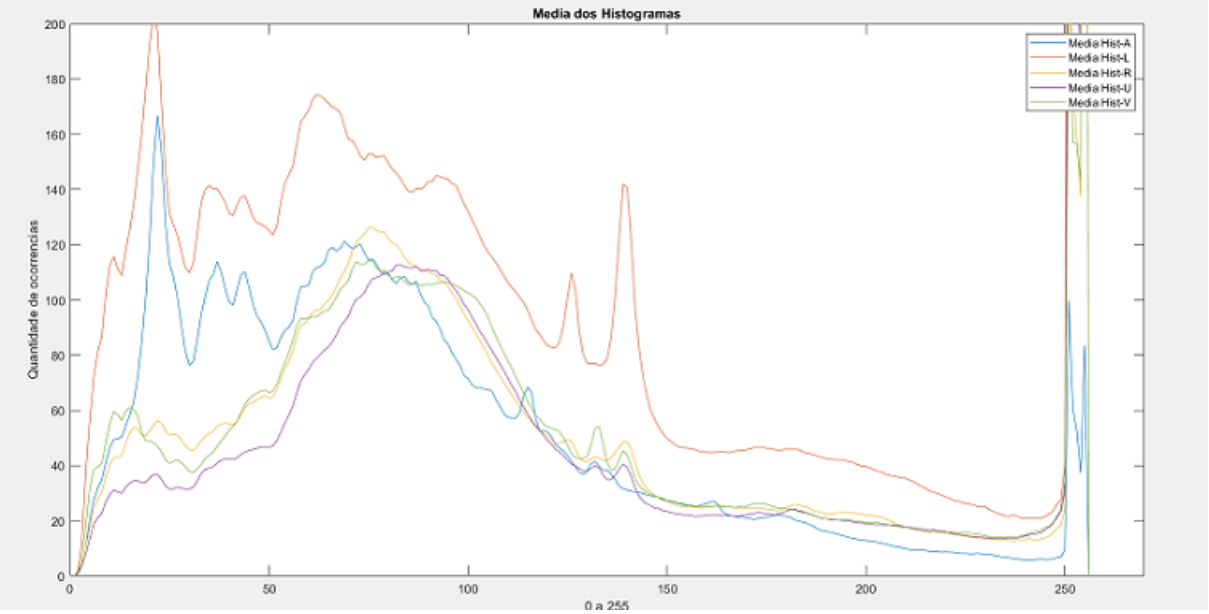
	Letras		
	R	U	V
Selecionada posição da mão	21%	15%	14%
Histograma Divergente	24%	16%	11%
Histograma Semelhante	20%	17%	15%

Fonte: Autor

4.3.3 Média dos histogramas

Na Figura 40 podemos observar que a curva da letra "L" se distancia bastante da outras curvas. Verificando na Tabela 16 na linha da letra "L": na coluna da letra "A" apresenta 0%, na coluna da letra "R" 2%, na coluna da letra "U" 1% e da letra letra "V" 1%.

Figura 40 – Histograma da letras "A", "L", "R", "U"e "V"



Fonte: Autor

4.4 Imagens adquiridas com o sensor kinect 360 e ONE

Nesta seção do trabalho serão apresentados os resultados das imagens que foram adquiridas com a utilização do sensor Kinect 360 e One. Devido algumas letras da linguagem LIBRAS serem representadas com movimentos, utilizamos somente 19 letras que representam as letras estáticas da linguagem.

A Tabela 21 mostra a quantidade de imagens que fazem parte do banco de dados. Temos as imagens em escala RGB que foram adquiridas tanto no sensor kinect 360 como no sensor Kinect One. Das imagens originais realizamos a segmentação e obtivemos um novo conjunto de imagens.

Tabela 21 – Quantidade de imagens por letras

	Kinect 360	Kinect ONE
	RGB, Depth e RGB segmentada	RGB, Depth e RGB segmentada
a	596	715
b	588	723
c	629	701
d	632	713
e	646	687
f	626	715
g	621	669
i	630	759
l	627	720
m	622	722
n	659	740
o	625	673
p	638	725
q	544	624
r	518	627
s	517	649
t	525	597
u	524	631
v	517	622
TOTAL	**11284**	**13012**

Fonte: Autor

4.4.1 Kinect 360

Foram usadas imagens RGB adquiridas com o sensor Kinect 360, sendo selecionamos as imagens originais em escala RGB, totalizando 11284 imagens. Na Tabela 22 podemos visualizar a matrix de confusão onde a acurácia obtida foi de 88%. É possível observar que a letra *a* tem acurácia mais alta (95%) e as letras *l, m* e *n* tem as acurácias mais baixas (81%). Na acurácia da letra *m* temos 6% confundida com a letra *n* e na acurácia da letra *n* temos 7% confundida com a letra *m*.

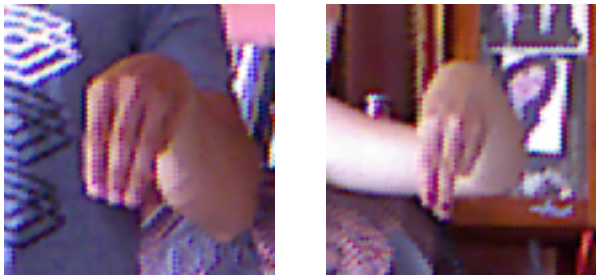
Nas imagens 41a e 41b com atores diferentes, podemos observar que as letras são bem parecidas comprovando a análise descrita anteriormente.

Tabela 22 – Matrix de confusão das imagens 360 RGB

Letras	a	b	c	d	e	f	g	i	l	m	n	o	p	q	r	s	t	u	v
a	0,95	0,00	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,00	0,01	0,00	0,00	0,00	0,00
b	0,00	0,91	0,00	0,01	0,01	0,00	0,00	0,01	0,00	0,01	0,00	0,00	0,01	0,00	0,01	0,01	0,00	0,01	0,01
c	0,01	0,00	0,87	0,02	0,02	0,00	0,00	0,00	0,01	0,01	0,01	0,02	0,01	0,00	0,00	0,00	0,00	0,00	0,00
d	0,00	0,01	0,01	0,84	0,01	0,02	0,01	0,02	0,01	0,01	0,01	0,03	0,00	0,01	0,01	0,01	0,00	0,01	0,00
e	0,01	0,02	0,00	0,01	0,88	0,01	0,02	0,01	0,00	0,01	0,00	0,00	0,01	0,00	0,00	0,01	0,00	0,00	0,00
f	0,00	0,00	0,01	0,01	0,00	0,86	0,03	0,01	0,00	0,01	0,01	0,01	0,00	0,00	0,00	0,00	0,04	0,01	0,00
g	0,01	0,01	0,00	0,00	0,01	0,01	0,88	0,02	0,02	0,00	0,01	0,00	0,00	0,00	0,00	0,01	0,01	0,00	0,00
i	0,00	0,01	0,00	0,01	0,01	0,01	0,03	0,85	0,02	0,00	0,01	0,00	0,00	0,01	0,00	0,01	0,01	0,01	0,00
l	0,00	0,00	0,01	0,01	0,01	0,01	0,02	0,03	0,81	0,00	0,02	0,00	0,02	0,01	0,00	0,01	0,02	0,01	0,02
m	0,01	0,00	0,01	0,00	0,01	0,00	0,00	0,01	0,00	0,81	0,06	0,01	0,00	0,04	0,00	0,01	0,00	0,01	0,01
n	0,01	0,00	0,01	0,00	0,00	0,00	0,01	0,00	0,01	0,07	0,81	0,01	0,02	0,03	0,00	0,01	0,00	0,01	0,00
o	0,01	0,00	0,01	0,02	0,01	0,01	0,00	0,01	0,01	0,00	0,01	0,88	0,01	0,01	0,00	0,01	0,00	0,00	0,00
p	0,00	0,00	0,00	0,00	0,01	0,00	0,00	0,00	0,01	0,01	0,01	0,01	0,93	0,01	0,00	0,00	0,00	0,00	0,00
q	0,00	0,00	0,00	0,00	0,01	0,00	0,00	0,00	0,00	0,01	0,01	0,00	0,01	0,93	0,00	0,00	0,01	0,00	0,00
r	0,00	0,01	0,00	0,01	0,00	0,00	0,02	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,89	0,01	0,00	0,02	0,01
s	0,00	0,00	0,00	0,00	0,01	0,00	0,01	0,00	0,01	0,00	0,01	0,01	0,00	0,00	0,01	0,93	0,01	0,00	0,00
t	0,00	0,00	0,00	0,00	0,00	0,02	0,00	0,01	0,01	0,00	0,00	0,00	0,00	0,01	0,00	0,00	0,92	0,01	0,00
u	0,00	0,01	0,00	0,00	0,00	0,01	0,01	0,01	0,01	0,01	0,01	0,00	0,00	0,00	0,05	0,01	0,01	0,77	0,08
v	0,00	0,00	0,00	0,00	0,00	0,01	0,00	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,02	0,00	0,00	0,03	0,91

Acurácia = 0,88

Fonte: Autor



(a) Letra m (b) Letra n

Figura 41 – Imagens RGB letra "m"e "n"

4.4.1.1 Segmentação da imagem RGB usando a imagem de profundidade segmentada

Para melhorarmos a acurácia do nosso classificador realizamos a seguinte segmentação. Utilizando a imagem de profundidade (Depth) aplicando metodologia descrita anteriormente conseguimos separar a mão do restante da imagem obtendo a Figura 42b. Esta imagem segmentada é multiplicada por cada componente da imagem RGB (R,G e B) 42a resultando na imagem segmentada Figura 42c.

Com este novo conjunto de imagens RGB segmentadas aplicamos o classificador e obtivemos uma nova matriz de confusão. Podemos observar que com as imagens segmentadas obtivemos uma melhora no nosso classificador. Sua acurácia foi 94% conforme pode ser visto na Tabela 23.

(a) Imagem Original RGB (b) Imagem profundidade segmentada (c) Imagem RBG segmentada

Figura 42 – Imagens RGB segmentadas utilizando imagem de profundidade segmentada

Tabela 23 – Matrix de confusão das imagens segmentadas - 360

Letras	a	b	c	d	e	f	g	i	l	m	n	o	p	q	r	s	t	u	v
a	0,95	0,00	0,00	0,00	0,01	0,00	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,02	0,00	0,00	0,00
b	0,00	0,97	0,00	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,00	0,00	0,00	0,00
c	0,00	0,00	0,96	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,00
d	0,00	0,00	0,01	0,87	0,02	0,00	0,01	0,01	0,00	0,00	0,00	0,05	0,00	0,00	0,02	0,01	0,00	0,00	0,00
e	0,01	0,01	0,00	0,00	0,93	0,00	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,02	0,00	0,00	0,00
f	0,00	0,01	0,00	0,00	0,00	0,87	0,02	0,00	0,00	0,00	0,00	0,00	0,01	0,00	0,00	0,01	0,06	0,00	0,00
g	0,02	0,00	0,00	0,00	0,00	0,00	0,93	0,00	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,00
i	0,01	0,00	0,00	0,00	0,01	0,00	0,00	0,96	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,00	0,00	0,00
l	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,00	0,98	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,01
m	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,81	0,15	0,00	0,00	0,02	0,00	0,00	0,00	0,00	0,00
n	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,13	0,85	0,00	0,00	0,01	0,00	0,00	0,00	0,00	0,00
o	0,00	0,00	0,01	0,02	0,00	0,01	0,00	0,00	0,00	0,00	0,00	0,95	0,00	0,00	0,00	0,00	0,00	0,00	0,00
p	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,99	0,00	0,00	0,00	0,00	0,00	0,00
q	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,00	0,00	0,99	0,00	0,00	0,00	0,00	0,00
r	0,00	0,00	0,00	0,01	0,00	0,00	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,93	0,00	0,00	0,03	0,00
s	0,00	0,00	0,00	0,00	0,01	0,00	0,00	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,97	0,00	0,00	0,00
t	0,00	0,00	0,00	0,00	0,00	0,03	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,95	0,00	0,00
u	0,00	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,04	0,00	0,01	0,91	0,01
v	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,98

Acurácia = 0,94

Fonte: Autor.

4.4.2 Kinect one

Utilizando as imagens adquiridas com o sensor Kinect One aplicamos o classificador obtendo a matrix de confusão 24. Nesta matrix a acurácia foi de 89%. As letras *a* e *b* tiveram a maior acurácia (98%) e a letra *n* teve a menor acurácia (77%).

Tabela 24 – Matrix de confusão das imagens RGB One

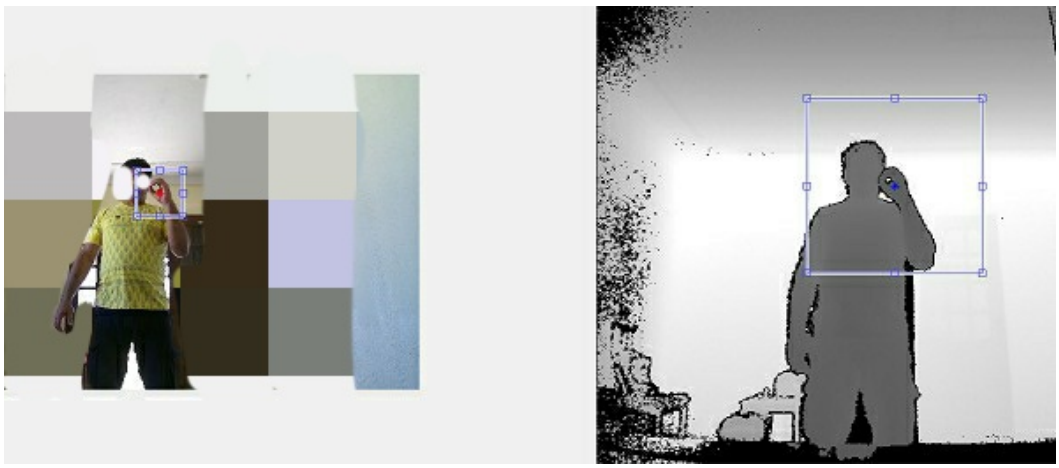
Letras	a	b	c	d	e	f	g	i	l	m	n	o	p	q	r	s	t	u	v
a	0,98	0,00	0,00	0,00	0,00	0,00	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
b	0,01	0,98	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
c	0,01	0,00	0,88	0,02	0,01	0,01	0,01	0,00	0,00	0,00	0,01	0,03	0,01	0,00	0,00	0,01	0,01	0,00	0,00
d	0,01	0,00	0,02	0,88	0,01	0,01	0,01	0,01	0,00	0,00	0,00	0,02	0,01	0,00	0,01	0,01	0,02	0,00	0,00
e	0,00	0,00	0,01	0,01	0,91	0,00	0,01	0,01	0,01	0,00	0,01	0,00	0,01	0,00	0,01	0,01	0,01	0,00	0,01
f	0,00	0,01	0,00	0,00	0,01	0,88	0,00	0,02	0,00	0,01	0,01	0,00	0,01	0,00	0,01	0,01	0,04	0,00	0,00
g	0,02	0,01	0,01	0,01	0,02	0,01	0,84	0,01	0,03	0,01	0,00	0,00	0,00	0,00	0,01	0,00	0,01	0,01	0,01
i	0,00	0,01	0,01	0,01	0,01	0,01	0,00	0,92	0,01	0,00	0,00	0,01	0,00	0,00	0,00	0,01	0,00	0,00	0,00
l	0,01	0,01	0,01	0,01	0,01	0,01	0,03	0,01	0,88	0,00	0,00	0,01	0,00	0,01	0,00	0,02	0,01	0,01	0,00
m	0,00	0,01	0,00	0,01	0,01	0,01	0,00	0,00	0,00	0,87	0,03	0,00	0,00	0,03	0,01	0,00	0,01	0,00	0,01
n	0,01	0,00	0,00	0,00	0,01	0,02	0,00	0,01	0,01	0,10	0,77	0,00	0,01	0,03	0,00	0,01	0,00	0,01	0,00
o	0,00	0,00	0,01	0,02	0,00	0,01	0,00	0,01	0,01	0,00	0,00	0,92	0,01	0,01	0,00	0,00	0,00	0,01	0,01
p	0,00	0,00	0,00	0,01	0,00	0,01	0,00	0,00	0,00	0,01	0,00	0,01	0,93	0,01	0,00	0,00	0,00	0,01	0,00
q	0,00	0,00	0,00	0,00	0,00	0,01	0,00	0,00	0,00	0,01	0,02	0,00	0,01	0,94	0,01	0,00	0,00	0,00	0,00
r	0,00	0,01	0,00	0,01	0,01	0,01	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,86	0,01	0,02	0,03	0,02
s	0,01	0,00	0,01	0,00	0,02	0,01	0,01	0,01	0,02	0,00	0,01	0,01	0,01	0,02	0,04	0,81	0,01	0,01	0,00
t	0,00	0,01	0,01	0,01	0,01	0,02	0,00	0,01	0,01	0,00	0,00	0,01	0,01	0,01	0,01	0,00	0,86	0,01	0,01
u	0,01	0,01	0,00	0,01	0,00	0,00	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,02	0,02	0,02	0,87	0,04
v	0,00	0,00	0,00	0,01	0,01	0,01	0,01	0,01	0,00	0,00	0,00	0,01	0,00	0,00	0,01	0,01	0,00	0,03	0,89

Acurácia = 0,89

Fonte: Autor.

Para melhorarmos a acurácia do classificador não podemos aplicar direto a metodologia utilizada com o sensor Kinect 360, devido a resoluções diferentes entre as imagens em RGB e profundidade, descritas na Tabela 2. Na Figura 43 podemos observar que os retângulos que determinam as ROI's, apesar de possuírem as mesmas dimensões (161x161), devido as resoluções diferentes a ROI captura regiões distintas do mesmo ator. No caso da imagem profundidade, esta capturou a cabeça do ator, enquanto que a RGB apenas uma parte da cabeça.

Figura 43 – Imagens One RGB e profundidade



Fonte: Autor.

Nosso ponto de partida é a imagem RGB, devemos criar uma forma de adquirir informações da imagem profundidade, devido esta ter o mapa de profundidade, o que nos ajuda na segmentação.

Observando a Figura 44, temos a necessidade de cortar parte da imagem profundidade (retângulo vermelho) de modo que as regiões sejam compatíveis entre as duas imagens.

Figura 44 – Imagens One RGB e profundidade com retângulo a ser recortado

Fonte: Autor.

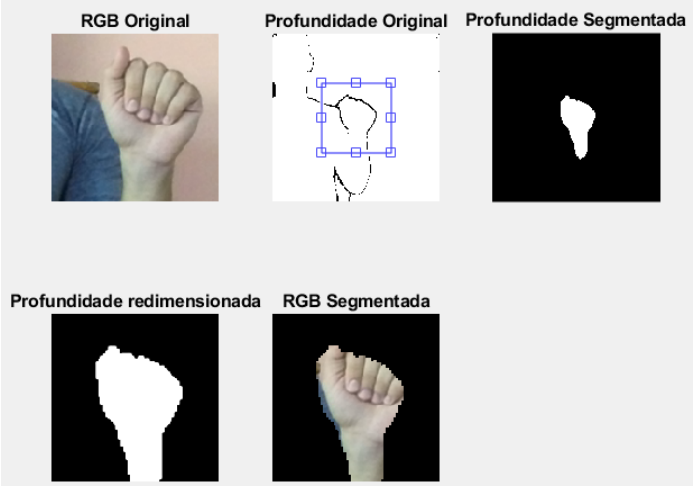
Na imagem de profundidade serão aplicados os seguintes processos, respectivamente:

- Aplicar na imagem profundidade Original a metodologia com base no histograma e obtemos a imagem profundidade segmentada;
- Na imagem profundidade segmentada recortamos a região determinada pela área de corte, porém esta imagem contém resoluções diferentes da RGB Original;
- Para resolvermos este problema é necessário aplicar um redimensionamento na imagem recortada para a mesma resolução da imagem RGB.

Estes passos realizados na imagens descritos acima são demonstrados na Figura 45.

Para encontramos um valor de área de corte da imagem profundidade foram testados diversos valores, sendo encontrado o valor 67x67 como aceitável. A Figura 45 mostra como a validação do valor foi realizada.

Figura 45 – Imagens testadas para segmentação



Fonte: Autor.

Após processo de segmentação aplicamos esse novo conjunto de imagens no classificador e obtivemos a matrix de confusão 25 onde a acurácia foi de 92%.

Tabela 25 – Matrix de confusão das imagens RGB One

Letras	a	b	c	d	e	f	g	i	l	m	n	o	p	q	r	s	t	u	v
a	0,96	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,00	0,00	0,00
b	0,00	0,95	0,00	0,00	0,01	0,00	0,00	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,00	0,02	0,00
c	0,00	0,00	0,93	0,00	0,01	0,01	0,00	0,01	0,00	0,00	0,00	0,02	0,00	0,00	0,00	0,01	0,01	0,00	0,00
d	0,00	0,00	0,00	0,93	0,00	0,00	0,00	0,02	0,00	0,00	0,00	0,02	0,00	0,00	0,02	0,01	0,00	0,00	0,00
e	0,01	0,01	0,00	0,01	0,92	0,00	0,00	0,01	0,00	0,00	0,00	0,01	0,00	0,00	0,01	0,02	0,00	0,00	0,00
f	0,00	0,01	0,00	0,01	0,01	0,81	0,01	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,12	0,00	0,01
g	0,01	0,00	0,00	0,00	0,00	0,00	0,95	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,01	0,00	0,01	0,00	0,00
i	0,00	0,00	0,00	0,00	0,00	0,01	0,00	0,96	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,00	0,00	0,00	0,01
l	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,00	0,98	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
m	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,00	0,84	0,13	0,00	0,00	0,02	0,00	0,00	0,00	0,00	0,00
n	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,16	0,77	0,00	0,00	0,04	0,00	0,01	0,00	0,00	0,00
o	0,00	0,00	0,03	0,01	0,01	0,00	0,00	0,01	0,00	0,00	0,00	0,91	0,00	0,00	0,00	0,02	0,00	0,00	0,00
p	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,00	0,00	0,00	0,00	0,99	0,00	0,00	0,00	0,00	0,00	0,00
q	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,02	0,00	0,00	0,96	0,00	0,00	0,00	0,00	0,00
r	0,00	0,00	0,00	0,01	0,01	0,00	0,02	0,01	0,00	0,00	0,01	0,00	0,00	0,00	0,88	0,00	0,00	0,05	0,01
s	0,01	0,01	0,01	0,00	0,02	0,00	0,01	0,01	0,00	0,00	0,00	0,02	0,00	0,00	0,01	0,90	0,00	0,01	0,00
t	0,00	0,01	0,00	0,01	0,00	0,03	0,00	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,93	0,00	0,01
u	0,00	0,01	0,00	0,02	0,00	0,01	0,00	0,01	0,00	0,00	0,00	0,01	0,00	0,00	0,08	0,01	0,01	0,84	0,02
v	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,00	0,00	0,01	0,96

Acurácia = 0,92

Fonte: Autor.

Tabela 26 – Acurácia das imagens com Kinect 360 e One

	Não segmentada Acurácia (%)	**Segmentada Acurácia (%)**
Kinect 360	88	94
Kinect One	89	92

Fonte: Autor

5 CONCLUSÃO

O reconhecimento de gestos tem sido um tema abordado por diversos pesquisadores. A interface homem máquina ainda carece de boas ferramentas para implementação. O reconhecimento de linguagem de sinais LIBRAS vem sido estudado e desenvolvidas técnicas obtidas em imagens em RGB com fundos uniformes para facilitar o processo de segmentação.

Na primeira parte deste trabalho utilizamos uma base de dados da ASL e estudamos o histograma das letras e realizamos algumas exclusões de imagens com esta base. Essas analises ficaram dependentes da avaliação humana.

Com relação a aquisição própria o trabalho conseguiu criar uma base de dados com imagens adquiridas em cidades diferentes e condições do ambiente diferentes. O número de atores e imagens é representativo e pode ser utilizado em trabalhos futuros por outros pesquisadores.

Apresentamos neste trabalho uma avaliação das possíveis soluções das etapas de segmentação. As imagens RGB's capturadas com o sensor Kinect 360 foram segmentadas a partir de suas imagens de profundidade pelo seus respectivos histogramas. Essa segmentação apresentou uma melhora na acurácia de 6%

Nas imagens RGB capturadas pelo sensor Kinect One utilizamos a mesma técnica de segmentação usada no Kinect 360, porém seus resultados foram inferiores aos valores obtidos nas imagens sem segmentação. A solução encontrada para resolução do problema descrita na seção dos resultados atendeu de forma satisfatória, onde sua acurácia teve uma melhora de 3%.

O trabalho realizou uma aquisição utilizando os dois Kinects de forma simultânea com computadores diferentes para cada sensor, onde cada sensor teve seu tempo de resposta. O sensor Kinect One obteve um melhor resultado em relação ao sensor 360 devido sua resolução.

Durante a aquisição das imagens verificamos experimentalmente que o sensor Kinect 360 tem uma maior rapidez no reconhecimento do usuário em relação ao Kinect One.

5.1 Trabalhos futuros

No trabalho não foi possível fazer o reconhecimento em tempo real das letras, o que seria de enorme valor para a comunidade acadêmica e em geral. Esta etapa ficará para trabalho futuros.

As imagens de profundidade foram utilizadas para criar uma segmentação na imagens em RGB. Estas imagens de profundidade utilizadas isoladamente são abordadas em outras literaturas, podendo ser objeto de estudo futuro.

Durante a aquisição tivemos uma situação particular que foi a alteração da distância do ator em relação ao Kinect, essa variação fez diferença no resultado, resultando numa oportunidade de estudos sobre o fato.

Referências

ALMEIDA, H. *principios sensoriamento remoto geoprocessamento*. 2013. Disponível em: <http://www.ebah.com.br/content/ABAAAgFJcAH/principios-sensoriamento-remoto-geoprocessamento?part=4>. Citado na página 28.

ALVARENGA, M. L. T.; CORREA, D. S. O.; OSÓRIO., F. S. Redes neurais artificiais no reconhecimento de gestos usando o kinect. *Computer on the beach*, 2012. Disponível em: <https://siaiap32.univali.br/seer/index.php/acotb/article/view/6602/3747>. Citado 2 vezes nas páginas 15 e 32.

ANDREOLA, R. *Support Vector Machines na classificação de imagens hiperesctrais*. Dissertação (mathesis) — Universidade federal do Rio Grande do Sul, Engenharia de Computação, 2009. Disponível em: <http://www.ufrgs.br/srm/ppgsr/publicacoes/Dissert_RafaelaAndreola.pdf>. Citado na página 30.

ANJO, M. dos S. *Avaliação das técnicas de segmentação, modelagem e classificação para o reconhecimento automático de gestos e proposta de uma solução para classificar gestos de libra em tempo real*. Dissertação (Ciência da computação) — Unifersidade Federal de São Carlos, 2013. Disponível em: <https://repositorio.ufscar.br/handle/ufscar/523>. Citado 4 vezes nas páginas 13, 14, 17 e 21.

BELUCO., A. *Classificação de imagens de sensoriamento remoto baseada em textura por redes neurais*. Dissertação (mathesis) — Universidade Federal do Rio Grande do Sul, 2002. Disponível em: <http://www.lume.ufrgs.br/handle/10183/6046>. Citado na página 27.

BRAZ, G. *Histogram of oriented gradients*. 2018. Disponível em: <http://nca.ufma.br/~geraldo/vc/13.hog.pdf>. Citado na página 35.

CHACON., G. T. *Aplicação de técnicas de processamento digital de imagens para a detecção de MARFEs no JET*. Dissertação (mathesis) — Centro Brasileiro De Pesquisas Físicas, Instrumentação Científica, 2012. Disponível em: <http://cbpfindex.cbpf.br/publication_pdfs/TeseChacon-072012.2012_07_26_15_12_47.pdf>. Citado 2 vezes nas páginas 29 e 31.

CORREIA, M. M. *Reconhecimento de elementos da língua gestual portuguesa com Kinect*. Dissertação (Engenharia de Eletrotécnica e de Computadores) — Universidade do Porto, 2013. Disponível em: <http://hdl.handle.net/10216/68032.Acesso.em.10.04.2017>. Citado 4 vezes nas páginas 14, 17, 21 e 24.

FILHO, O. M.; NETO., H. V. *Processamento digital de imagens*. [S.l.]: Brasport, 1999. Citado 7 vezes nas páginas 13, 14, 15, 22, 23, 24 e 25.

FUNCAP. *II Seminário de Educação de Surdos e Libras*. 2015. Disponível em: <https://www.funcap.ce.gov.br/2015/08/24/seminario-de-educacao-de-surdos-e-libras-abre-inscricoes/>. Citado na página 49.

GONZALES, R. C.; WOODS, R. E.; EDDINS., S. L. *Digital image processing*. [S.l.]: Prentice hall Upper Saddle River, NJ:, 2004. Citado na página 22.

INES. *Dicionário da lingua brasileira de sinais*. 2005. Disponível em: <http://www.ines.gov.br/dicionario-de-libras/main_site/libras.htm>. Citado na página 26.

JUNIOR., J. P. da S. *Alinhamento de imagens de profundidade com aplicação no reconhecimento da língua de sinais*. Dissertação (Informática) — Universidade de Brasília, 2014. Disponível em: <http://repositorio.unb.br/handle/10482/16978>. Citado 3 vezes nas páginas 12, 14 e 16.

LAVRENKO, V. *K-means clustering: how it works*. 2014. Disponível em: <https://www.youtube.com/watch?v=_aWzGGNrcic>. Citado 2 vezes nas páginas 28 e 29.

MATHWORKS. *Extract histogram of oriented gradients HOG features*. 2013. Disponível em: <https://www.mathworks.com/help/vision/ref/extracthogfeatures.html>. Citado 2 vezes nas páginas 35 e 36.

MATHWORKS. *Image Classification with Bag of Visual Words*. 2019. Disponível em: <https://www.mathworks.com/help/vision/ug/image-classification-with-bag-of-visual-words.html>. Citado na página 36.

MATSUNAGA, V. Y. *Curso de redes neurais utilizando o matlab*. 2012. Disponível em: <http://www.muriloleal.com.br/visao/repositorio/centec/eai/ia/REDES%20NEURAIS%20-%20APOSTILA.pdf>. Citado 2 vezes nas páginas 33 e 34.

MENDONÇA., V. G. de. *Método para classificação de um conjunto de gestos usando Kinect*. Dissertação (Mestrado em informatica) — Pontifícia Universidade Católica do Paraná, 2013. Disponível em: <https://www.ppgia.pucpr.br/pt/arquivos/mestrado/dissertacoes/2013/vinicius-godoy-VF.pdf>. Citado 3 vezes nas páginas 12, 16 e 17.

MENESES, P. R.; ALMEIDA, T. *Introdução ao processamento de imagens de sensoriamento remoto*. Unb, 2012. Disponível em: <http://www.cnpq.br/documents/10157/56b578c4-0fd5-4b9f-b82a-e9693e4f69d8>. Citado 4 vezes nas páginas 21, 24, 25 e 27.

MONTEIRO, C. H. de A. et al. Um sistema de baixo custo para reconhecimento de gestos em libras utilizando visão computacional. *SIMPÓSIO BRASILEIRO DE TELECOMUNICAÇÕES*, 2016. Disponível em: <http://www.sbrt.org.br/sbrt2016/anais/ST11/1570279225.pdf>. Citado 3 vezes nas páginas 15, 16 e 26.

NASCIMENTO, R. F. F. et al. O algoritmo support vector machines (svm): avaliação da separação ótima de classes em imagem ccd-cbers-2. *Anais XIV Simpósio Brasileiro de Sensoriamento Remoto*, 2009. Disponível em: <http://marte.sid.inpe.br/col/dpi.inpe.br/sbsr@80/2008/10.20.10.59/doc/2079-2086.pdf>. Citado na página 30.

OPENCV. *OpenCv Tutorials*. 2018. Disponível em: <https://docs.opencv.org/master/d9/df8/tutorial_root.html>. Nenhuma citação no texto.

PAVAN, A. R.; CAZHURRIRO, J.; MODESTO., F. Reconhecimento de gestos com segmentação de imagens dinâmicas aplicadas a libras. *Anuario da produção de iniciação científica discente*, v. 13, n. 20, 2010. Disponível em: <http://repositorio.pgsskroton.com.br/bitstream/123456789/1240/1/artigo%2023.pdf>. Citado 4 vezes nas páginas 14, 15, 21 e 32.

PEDROSA, G. V. *Caracterização e recuperação de imagens usando dicionários visuais semanticamente enriquecidos*. Tese (Instituto de Ciências Matemáticas e de Computação) — USP - São Carlos, 2015. Disponível em: <http://www.teses.usp.br/teses/disponiveis/55/55134/tde-19122015-120703/es.php>. Citado na página 36.

QIDWAI, U.; CHEN, C.-h. *Digital image processing: an algorithmic approach with MATLAB*. [S.l.]: Chapman and Hall/CRC, 2009. Unico. ISBN 1138115185. Citado na página 15.

RODRíGUEZ, K. C. O. *Reconhecimento de Sinais Estáticos a partir de Infomação RGB-D usando um Descritor Kernel*. Dissertação (Ciência da computação) — Universidade Federal de Ouro Preto, 2014. Disponível em: <https://www.repositorio.ufop.br/bitstream/123456789/4213/1/DISSERTA%C3%87%C3%83O_ReconhecimentoSinaisEst%C3%A1ticos.pdf>. Citado na página 36.

SOARES, T. B. de M. M. J.; RAIA., F. Utilizando o kinect como auxílio sensorial para portadores de deficiências visuais. *COBENGE*, 2014. Disponível em: <http://198.136.59.239/~abengeorg/cobenge-2014/Artigos/129269.pdf>. Citado 2 vezes nas páginas 15 e 20.