

**UNIVERSIDADE DE SÃO PAULO**  
**ESCOLA DE ENGENHARIA DE SÃO CARLOS**  
**DEPTO. DE ENGENHARIA ELÉTRICA E DE**  
**COMPUTAÇÃO**

**SEL0630 - Aplicação de Microprocessadores II**

**Interpretador de Sinais aplicado à Língua**  
**Brasileira de Sinais - LIBRAS**

**Autor:** Kollins Gabriel Lima, nº. USP 9012931

**Autor:** Nome Aluno, nº. USP 0000000

**Orientador:** Prof. Dr. Evandro Luis Linhari Rodrigues

São Carlos

2017



# Resumo

Texto em **UM PARÁGRAFO** apenas - deve conter tudo resumidamente (introdução, método(s), resultados e conclusões), de tal forma que seja possível compreender a proposta e o que foi alcançado.

Palavras-Chave: palavra1, palavra2, palavra3, palavra4, palavra5.



# Abstract

[illegible]

**Keywords:** keyword1, keyword2, keyword3, keyword4, keyword5.



# Lista de Figuras

3.1	Método utilizado. . . . .	22
3.2	Etapas do pré-processamento. . . . .	23
3.3	Resultado do pré-processamento. . . . .	24
3.4	Diagrama simplificado do sistema. . . . .	25
4.2	Diagrama simplificado do sistema. . . . .	27
4.1	Diagrama simplificado do sistema. . . . .	28





# Lista de Tabelas

(Se houver...)



## Siglas (Se houver...)

MVC	<i>Model-View-Controller</i> - Modelo-Visão-Controlador
POO	Programação Orientada a Objetos
UI	<i>User Interface</i> - Interface do Usuário
UML	<i>Unified Modeling Language</i> - Linguagem de Modelagem Unificada



# Sumário

<b>1</b>	<b>Introdução</b>	<b>15</b>
1.1	Objetivo . . . . .	16
1.2	Justificativa . . . . .	16
1.3	Organização do Trabalho . . . . .	16
<b>2</b>	<b>Fundamentação Teórica</b>	<b>17</b>
2.1	Segmentação . . . . .	17
2.1.1	Detecção de Pele . . . . .	17
2.1.2	Detecção de Borda . . . . .	18
2.1.3	Detecção de Face . . . . .	18
2.2	<i>Hash</i> . . . . .	18
2.3	Classificação . . . . .	19
2.3.1	Classificador k-NN . . . . .	19
<b>3</b>	<b>Materiais e Métodos</b>	<b>21</b>
3.1	Material . . . . .	21
3.2	Métodos . . . . .	22
3.2.1	Captura da Imagem . . . . .	23
3.2.2	Pré-Processamento . . . . .	23
3.2.3	<i>Hash</i> . . . . .	24
3.2.4	Classificação . . . . .	24
<b>4</b>	<b>Resultados e Discussões</b>	<b>27</b>
<b>5</b>	<b>Conclusão ou Conclusões</b>	<b>31</b>
	<b>Referências</b>	<b>31</b>

<b>Apêndices</b>	<b>35</b>
<b>Apêndice A Cuidados e orientações para a elaboração de texto</b>	<b>35</b>
<b>Apêndice B Apresentação do TCC</b>	<b>39</b>
<b>Apêndice C Monografia Parcial do TCC</b>	<b>41</b>
<b>Anexos</b>	<b>41</b>
<b>I Aquilo que não é de sua autoria</b>	<b>43</b>
<b>II Aquilo que não é de sua autoria mas você julga importante colocar aqui</b>	<b>45</b>

# Capítulo 1

## Introdução

A Língua Brasileira de Sinais (Libras) é reconhecida oficialmente desde 2002 por meio da lei Nº 10.436 (regulamentada pelo decreto Nº 5.626 de 2005). Este marco representa um grande avanço para a comunidade surda, que possuem agora uma língua própria e direitos garantidos por lei, como expresso nos artigos 2º, 3º e 4º das obrigações do governo em divulgar a Libras e garantir o atendimento e tratamento à comunidade surda [1].

Segundo o Portal Brasil, existem mais de 9,7 milhões de surdos no país (dados do Censo de 2010), no entanto, a acessibilidade para essas pessoas ainda é um desafio devido a falta de intérpretes [2]. No intuito de diminuir as barreiras de comunicação, vários aplicativos tem surgido com a proposta de traduzir sites, textos e até mesmo voz para a Libras, como é o caso do ProDeaf e do Hand Talk apenas para citar alguns. No entanto, o processo inverso de tradução, ou seja, de Libras para o português, não é uma funcionalidade muito comum nestes aplicativos.

É vasta a quantidade de estudos envolvendo o reconhecimento de gestos aplicadas à línguas de sinais. Na literatura, encontramos dois métodos principais para a leitura dos sinais: abordagem visual, baseada em visão computacional, ou por meio de dispositivos eletromecânicos, utilizando luvas, sensores, etc. Embora a segunda abordagem facilite o processo de aquisição do sinal, ela requer o uso de um hardware adicional que pode ser desconfortável e custoso. A primeira abordagem, entretanto, é mais simples para uso no dia-a-dia, mas requer um maior pré-processamento para identificação e reconhecimento das mãos [3].

Também é vasta a coleção de aplicativos disponíveis para traduzir o português para Libras, dois grandes exemplos a serem citados são o *Hand Talk* e o *ProDeaf*, duas soluções nacionais premiadas internacionalmente.

No entanto, sistemas capazes de fazer o processo inverso de tradução e que estão disponíveis à sociedade ainda são escassos. Em 2016, o projeto *SignAloud*, desenvolvido por Navid Azodi e

Thomas Pryor, ambos estudantes da Universidade de Washington, ganharam o prêmio Lemelson, fornecido pelo Instituto de Tecnologia de Massachusetts, pelo projeto de reconhecimento da língua ASL (*American Sign Language* ou Língua de Sinais Americana). O projeto é baseado em uma luva equipada com placas de arduino e acelerômetros para o reconhecimento de sinais e ainda não possui uma versão comercial.

Um projeto semelhante, apresentado em Julho de 2017 por engenheiros da Universidade da Califórnia em San Diego, é o *The Language of Glove*, com a mesma proposta de reconhecimento de sinais ASL, mas utilizando um hardware de baixo custo (menos de 100 dólares, segundo os autores).

Aqui no Brasil, foi lançado em junho de 2017, projeto Giulia para Android. Idealizado pelo Professor Manuel Cardoso da Ufam (Universidade Federal do Amazonas), o projeto utiliza os próprios sensores do *smartphone*, que fica preso ao braço do usuário, para fazer o reconhecimento dos sinais, eliminando a necessidade de hardwares adicionais.

A grande motivação deste trabalho é atuar de forma a fornecer ferramentas para que um usuário de Libras possa interagir com pessoas que não tem conhecimento da língua de sinais. A proposta inicial é utilizar o método de visão computacional para a identificação de gestos estáticos, focando no reconhecimento dos números de 0 a 7.

## 1.1 Objetivo

Este trabalho tem como objetivo principal o desenvolvimento de um aplicativo para interpretação de sinais em Libras, fornecendo uma resposta textual ao sinal identificado.

## 1.2 Justificativa

Este projeto se justifica por atuar de forma a buscar alternativas à falta de intérpretes disponíveis aos usuários de Libras, não com o intuito de substituí-los, mas na tentativa de aumentar a autonomia dos deficientes auditivos em duas tarefas diárias, e agregar valor tecnológico e inclusivo ao Brasil.

## 1.3 Organização do Trabalho

Este trabalho está distribuído em XXX capítulos, incluindo esta introdução, dispostos conforme a descrição que segue:

- Capítulo 2: Descreve .....
- Capítulo 3: Discorre sobre .....
- Capítulo 4: Apresenta .....



## Capítulo 2

# Fundamentação Teórica

### 2.1 Segmentação

Ao se capturar uma imagem para processamento, é muitas vezes conveniente separar regiões de interesse, eliminando elementos alheios. Para este projeto, estamos interessados em obter uma imagem das mãos com o maior nível de detalhe possível.

Na tentativa de isolar as mãos em uma imagem foram adotadas 3 técnicas:

- Detecção de pele
- Detecção de bordas
- Detecção de face

#### 2.1.1 Detecção de Pele

Uma técnica utilizada para detecção de pele é a baseada em cores, ou seja, é considerada pele todos os pixels que estão dentro de uma faixa de cores. Matematicamente, seja a imagem  $F(x,y)$  e dois limitantes  $\tau_1$  e  $\tau_2$ , a detecção de pele é feita conforme a equação 2.1.

$$F(x,y) = \begin{cases} F(x,y), & \text{se } \tau_1 > F(x,y) > \tau_2 \\ 0, & \text{caso contrário} \end{cases} \quad (2.1)$$

Nos trabalhos consultados, não houve unanimidade quanto à melhor opção de espaço de cores para detecção de pele. Em [4], é utilizado o espaço de cores YCbCr, o mesmo utilizado por [5]. Em [6] é utilizado o espaço de cores YUV enquanto [7] faz o reconhecimento de pele no espaço RGB.

Experimentalmente, o melhor resultado foi obtido com o uso do espaço de cor HSV, como feito por [8], que também aponta as limitações do uso da técnica baseada em cores devido a grande variação

de tonalidade de peles.

### 2.1.2 Detecção de Borda

A detecção de borda realça os limites das regiões, baseando-se na variação de luminosidade [9]. Este processo se mostra necessário pois realiza a binarização da imagem, evidenciando o elemento de interesse (mãos) e diminuindo a quantidade de características presentes na imagem, o que facilita a classificação posteriormente [9].

### 2.1.3 Detecção de Face

A detecção de face é uma técnica que visa identificar a existência ou não de uma face humana na imagem. Difere do reconhecimento de face, já que não se deseja identificar a pessoa, mas apenas identificar que existe uma pessoa (face).

Uma vez que se deseja isolar as mãos, é preciso fazer a identificação e a remoção da face, já que a detecção de pele destaca estas duas regiões [5].

Estas 3 etapas mostradas constituem uma fase de pré-processamento da imagem, com o intuito de identificar e isolar as mãos do ambiente.

## 2.2 Hash

Uma função *hash* é uma função unidirecional que mapeia dados de entrada em uma faixa limitada de valores. No presente trabalho é utilizada a função *hash* para mapear uma imagem em um valor numérico de 256-bits.

Para fazer a transformação, é utilizado um algoritmo de *Average Hash* seu funcionamento é descrito a seguir [10] para a criação de um *hash* de 64-bits:

- A imagem de entrada é reduzida à 64 pixels (8x8) de modo a eliminar detalhes e componentes de altas frequências.
- Conversão para escala de cinza.
- Cálculo do valor médio (nível de cinza médio).
- A imagem é binarizada seguindo o critério: se o pixel estiver acima da média ele é setado (recebe 255), do contrário recebe o valor 0.
- A leitura da imagem (da esquerda para direita, de cima para baixo) gera um código binário, que representa o hash desejado.

## 2.3 Classificação

Para fazer o reconhecimento das imagens é preciso fazer uso de um sistema de classificação. Este sistema faz a identificação da imagem com base em um conjunto de treinamento previamente fornecido.

Vários são os métodos para fazer a classificação. Em [4] é utilizado máquina de vetores de suporte ou SVM (*Support Vector Machine*); classificador de Bayes é utilizado em [6] e em [7] utiliza-se Modelo oculto de Markov. Para simplificar o desenvolvimento, no entanto, foi adotado um classificador simples para este projeto, o k-NN (k-nearest neighbors).

### 2.3.1 Classificador k-NN

O classificador k-NN (*k-Nearest Neighbors*) verifica a distância do elemento de entrada para cada elemento no conjunto de treinamento e faz a classificação baseado nos k elementos mais próximos. A medida de distância deve ser definida com base nos elementos a serem classificados.

O k-NN suporta naturalmente uma aprendizagem incremental e utiliza uma abordagem *lazy*, ou seja, o algoritmo de aprendizagem retém informações de treinamento e apenas constroi um modelo ao receber uma entrada a ser classificada [11].



## Capítulo 3

# Materiais e Métodos

### 3.1 Material

Com a intenção de solucionar um problema do mundo real e facilitar vidas no que diz respeito à acessibilidade, de maneira a realmente otimizar de alguma forma o mundo das pessoas envolvidas, surgiu a idéia de desenvolver um sistema para que quem tenha deficiência auditiva possa se comunicar com outros que desconhecem a linguagem para surdos mudos, já que esta é pouco difundida e ensinada, trazendo desta forma melhorias para os deficientes em sua desenvoltura pessoal, vida profissional e educação, além de incentivar a aprendizagem da Libras.

O Projeto base tem por objetivo estruturar e desenvolver um sistema de interpretação dos números do alfabeto da Libras de 0 a 7.

Sendo assim, o Projeto principal consiste em um aplicativo Android conectado a um sistema embarcado em uma Raspberry Pi com Raspian rodando em rede e com comunicação Web para um website para treinamento de rotinas da Linguagem. Ou seja, generalizando a divisão do projeto, tem-se os seguintes materiais envolvidos:

- Android Studio: Para a criação do aplicativo, o qual processa as imagens da câmera do Smartphone para transferi-lás à Raspberry e aparecendo concomitantemente no website implementado
- Raspberry Pi: Pequeno microcomputador, embarcado com uma distribuição Linux e programado para as necessidades do projeto de interpretar a linguagem de Libras.
- Raspian: Distribuição variante do Debian do Linux baseada na arquitetura ARM do hardware da Raspberry Pi.
- Python: Linguagem de programação interpretada de alto nível utilizada durante toda execução do projeto, tanto para rotinas internas de classificação, quanto para criação do website.

- Flask: Framework pequena e poderosa para Python utilizada na criação do sistema Web.
- OpenCV: Significando Open Source Computer Vision Library, é uma biblioteca multiplataforma utilizada no aplicativo Android afim de processar as imagens obtidas na câmera.
- Materilize: Framework FRont-End com uma interface gráfica mais agradável, utilizada em nosso sistema web de treinamento de símbolos.
- MongoDB: Modelo de banco de dados NoSQL que facilita sua criação, servindo como armazenamento das imagens dos símbolos das Libras para posteriores comparações e classificações.

O sistema e suas comunicações, em resumo, funcionam da seguinte maneira: O aplicativo Android é responsável por adquirir a imagem via câmera do *Smartphone*, conta em sua implementação com OpenCV (Open Source Computer Vision Library) para interpretar as imagens, detectando bordas, pele e face, envia então essas imagens de modo simultâneo para o sistema rodando na Raspiberry

## 3.2 Métodos

A metodologia utilizada para realizar este trabalho baseia-se nas metodologias utilizadas em [4], [5] e [7]. Basicamente o processo se divide conforme mostra a figura 3.1.

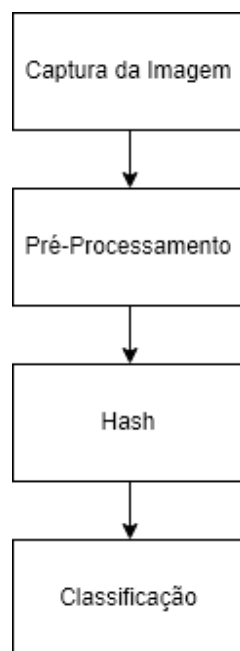


Figura 3.1: Método utilizado.

### 3.2.1 Captura da Imagem

A captura da imagem é feita pela câmera do aparelho Android. Não foi imposta nenhuma restrição quanto às características da câmera para este projeto.

### 3.2.2 Pré-Processamento

O pré-processamento é realizado com auxílio da biblioteca *OpenCV* e consiste em tratar a imagem capturada pela câmera de modo isolar as características desejadas (neste caso, busca-se isolar a região das mãos).

O processo de pré-processamento é mostrado na figura 3.2. São realizadas 3 etapas de maneira independente:

- Detecção de bordas: feita por meio da função *Imgproc.Canny*.
- Detecção de face: utiliza o *CascadeClassifier* da biblioteca *OpenCV* para fazer a detecção da região quadrada que contém uma face. Esta região é então preenchida (preto) de modo a eliminar a face da imagem.
- Detecção de pele: detecta regiões dentro de uma faixa de cores (utilizando o espaço HSV). O ajuste para detecção de pele foi feito manualmente.

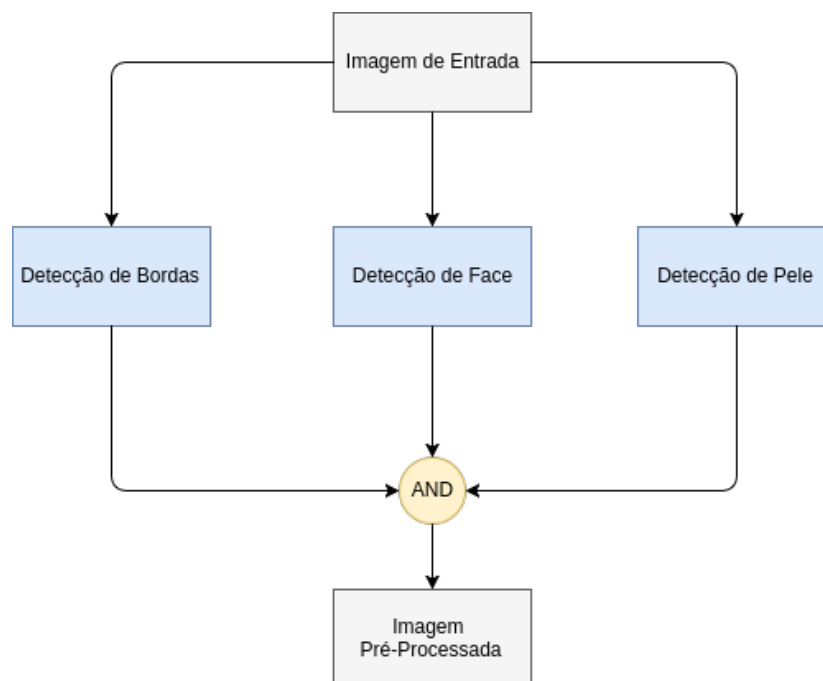


Figura 3.2: Etapas do pré-processamento.

Ao final, o resultado dos 3 processos são combinados em uma única imagem. O resultado é uma imagem como mostrado na figura 3.3, onde apenas o contorno das mãos fica visível na tela. Esta imagem é convertida para "jpg" e enviada pela rede à Raspberry onde ocorrerá a segunda parte do processo.

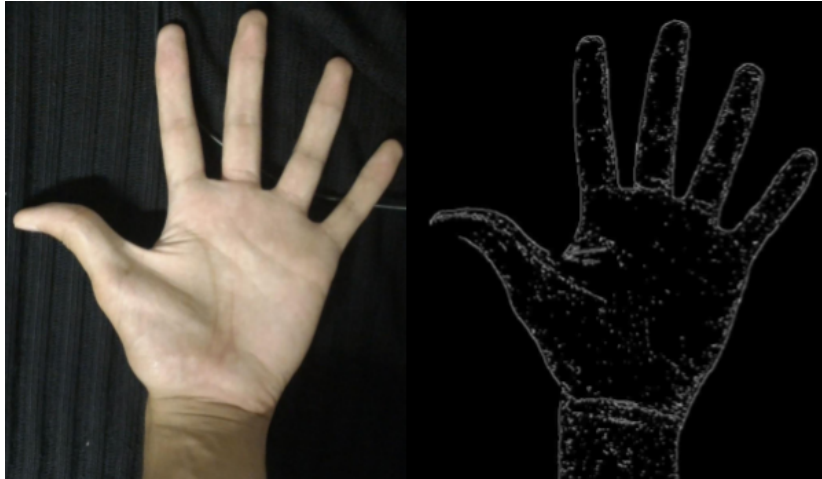


Figura 3.3: Resultado do pré-processamento.

### 3.2.3 Hash

A imagem recebida passa pelo processo de *Average Hash*, disponibilizado pela biblioteca *ImageHash* do Python. Este processo é feito para facilitar a comparação das imagens pois, ao final do processo, as imagens serão representadas por números, que podem ser facilmente armazenadas na base de dados e comparada com operações aritméticas simples.

Experimentalmente, chegou-se ao valor de 256-bits para o tamanho do *hash*, abaixo deste valor não era possível diferenciar as imagens com o classificador utilizado.

### 3.2.4 Classificação

Por meio da interface web desenvolvida, é possível visualizar cada imagem recebida pela Raspberry e fazer o treinamento do banco de dados relacionando a imagem visualizada ao seu número correspondente.

Ao mesmo tempo, em segundo plano, o classificador *K-NN* executa continuamente a classificação de cada imagem recebida. Foram utilizados os 3 resultados melhores classificados (3 vizinhos). Este valor também foi decidido experimentalmente de forma a não consumir muito tempo no processo de classificação.

Antes de enviar o resultado novamente ao Android, foi aplicado um *threshold* buscando diminuir a taxa de erro. Desta forma, cada novo resultado obtido do classificador só é enviado pela rede como



resultado final se for, pelo menos, 10% diferente do resultado enviado anteriormente. O valor de 10% foi obtido verificando como o classificador se comporta quando não há mudança no símbolo de entrada(qual o erro nesta condição).

A imagem 3.4 apresenta um diagrama simplificado do funcionamento do sistema como um todo.

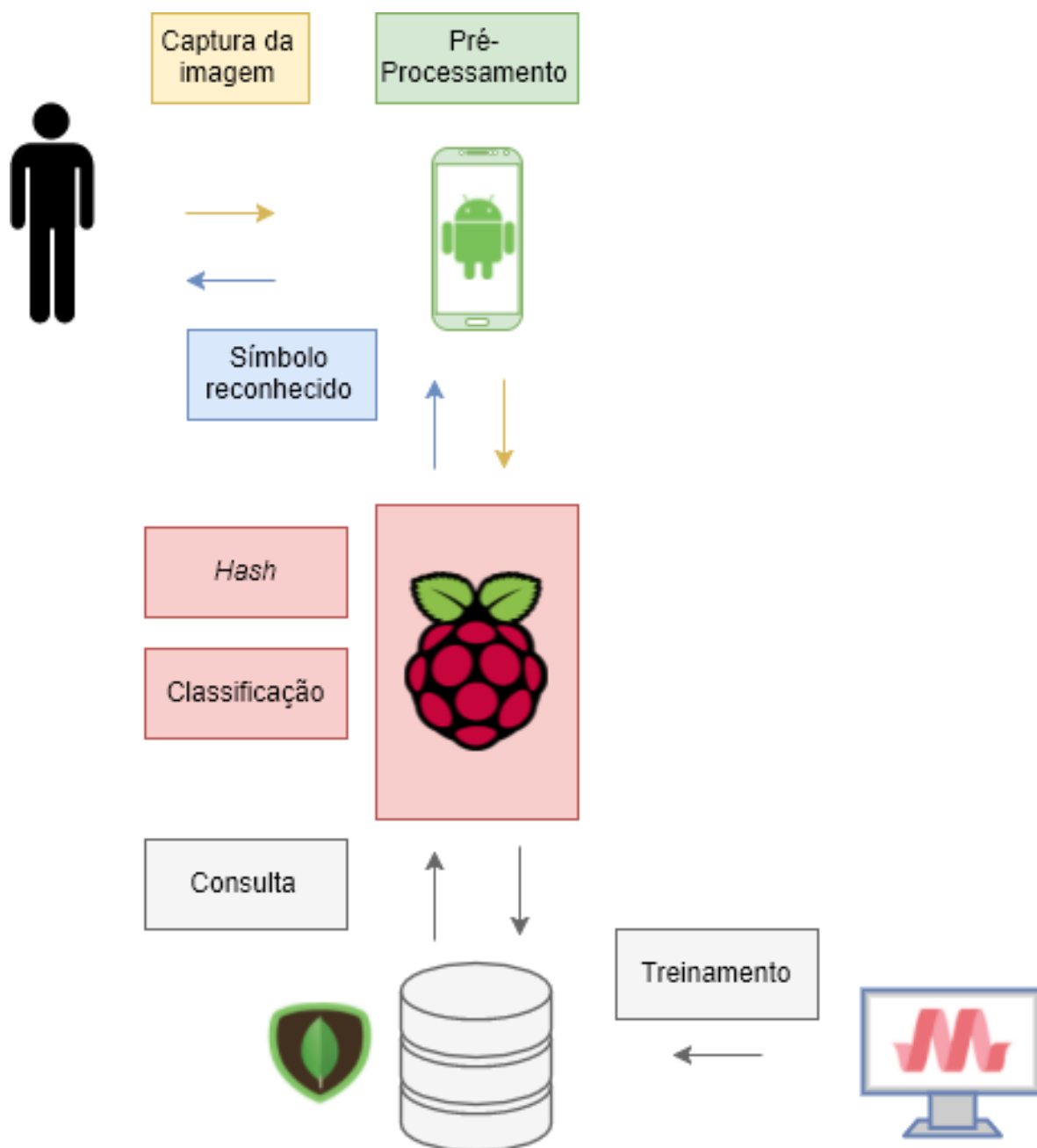


Figura 3.4: Diagrama simplificado do sistema.



## Capítulo 4

# Resultados e Discussões

Os símbolos a serem classificados são mostrados na figura 4.1.

O sistema funciona com baixas taxas de erro quando trabalha com poucos símbolos, no entanto o processo de classificação se torna impreciso conforme a quantidade de símbolos diferentes aumenta. A figura 4.2 mostra a taxa de erro obtida em função da quantidade de símbolos diferentes que foram adicionados ao banco de dados.

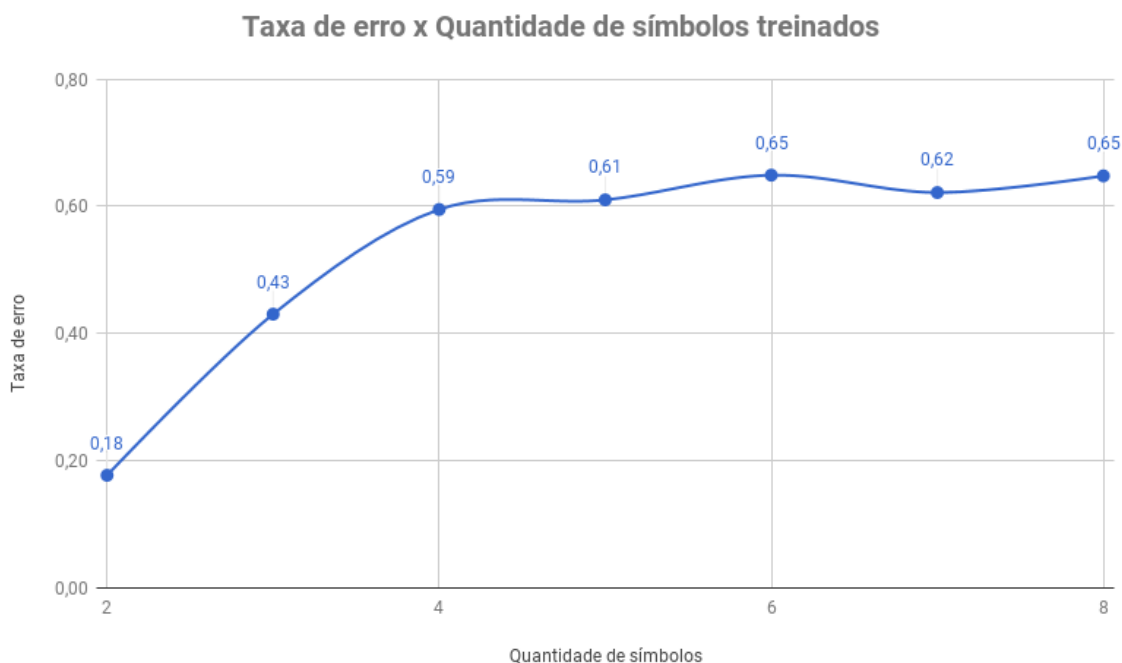


Figura 4.2: Diagrama simplificado do sistema.

Para realizar os testes, o banco de dados foi treinado com a mesma quantidade de exemplos para cada símbolo (3 exemplos) e foi dada como entrada a mesma quantidade de instâncias de cada símbolo

	0
	1
	2
	3
	4
	5
	6
	7

Figura 4.1: Símbolos propostos para o sistema de classificação.

(3 instâncias para cada símbolo treinado). O gráfico representa a taxa de desequilíbrio entre o esperado e o obtido.

Como mostrado na figura 4.2, acima de 2 símbolos o sistema é caótico, com baixas taxas de acerto. Esse resultado indica duas possibilidades para que o sistema não funcione como o esperado:

1. O processo de *hash* elimina muitas características da imagem de forma que diferenciar muitas imagens se torna uma tarefa complicada.
2. O classificador utilizado (*K-NN*) não é bom o suficiente para classificar muitos símbolos.

Muito embora o classificador *K-NN* seja bastante simples se comparado às técnicas utilizadas nos trabalhos citados anteriormente, acredita-se que a transformação *hash* seja a principal responsável pelos resultados obtidos.



## **Capítulo 5**

# **Conclusão ou Conclusões**

Conclusões do trabalho de conclusão de curso.

### **Trabalhos futuros**

Isso é para a Monografia Final de defesa.....





## Referências

- [1] Claudio Alves Benassi. Da lei n. 10.436 de 24 de abril de 2002, 2014.
- [2] Portal Brasil (Org.). Apesar de avanços, surdos ainda enfrentam barreiras de acessibilidade. Disponível em: <<http://www.brasil.gov.br/cidadania-e-justica/2016/09/apesar-de-avancos-surdos-ainda-enfrentam-barreiras-de-acessibilidade>>. Acesso em: 08 out. 2017, 2016.
- [3] Luciano A. Digiampietri Beatriz Teodoro. Desenvolvimento de um sistema de reconhecimento automático de língua brasileira de sinais. Disponível em: <http://www.each.usp.br/digiampietri/bibtex/TeodoroEDigiampietri2014a.pdf>. Acesso em: 22 out. 2017, 2014.
- [4] João D. S. Almeida Ruberth A. A. Barros, Aitan V. Pontes. Reconhecimento de linguagem de sinais: aplicação em libras, 2014.
- [5] Hyotaek Lim Hui-Shyong Yeo, Byung-Gook Lee. Hand tracking and gesture recognition system for human-computer interaction using low-cost hardware, 2013.
- [6] Yanyu Niu Xiaotang Wen. A method for hand gesture recognition based on morphology and fingertip angle, 2010.
- [7] Chung-Lin Huang Feng-Sheng Chen, Chih-Ming Fu. Hand gesture recognition using a real-time tracking method and hidden markov models, 2003.
- [8] Adrian Rosebrock. Skin detection: A step-by-step example using python and opencv. Disponível em: <<https://www.pyimagesearch.com/2014/08/18/skin-detection-step-step-example-using-python-opencv/>>. Acesso em: 15 out. 2017, 2014.
- [9]
- [10] Neal Krawetz. Looks like it. Disponível em: <http://www.hackerfactor.com/blog/index.php/?archives/432-Looks-Like-It.html>. Acesso em: 12 nov. 2017, 2011.

- [11] Solange Oliveira Rezende Rafael Geraldeli Rossi. Aprendendo com os vizinhos, 2017.

## Apêndice A

# Cuidados e orientações para a elaboração de texto

### Dicas para a redação de uma boa monografia de TCC

Observe as diretrizes no site do Depto.

[http://143.107.182.35/sel/files\\_EE/tcc\\_-\\_diretrizes\\_EESC\\_v\\_2010.pdf](http://143.107.182.35/sel/files_EE/tcc_-_diretrizes_EESC_v_2010.pdf)

Veja no Portal de Livros Abertos da USP as mais novas versões das Diretrizes para apresentação de dissertações e teses da USP.

- Parte I (ABNT) - <http://dx.doi.org/10.11606/9788573140606>
- Parte II (APA) - <http://dx.doi.org/10.11606/9788573140576>
- Parte III (ISO) - <http://dx.doi.org/10.11606/9788573140590>
- Parte IV (Vancouver) - <http://dx.doi.org/10.11606/9788573140569>

Observe os elementos pré-textuais neste documento....tem uma sequência a ser seguida (Capa, contracapa, Ficha catalográfica para a versão final, Listas de Figuras, Tabelas e Símbolos/Abreviaturas).

### Resumo/abstract

Texto em um parágrafo apenas. Deve conter tudo resumidamente (introdução, método(s), resultados e conclusões), de tal forma que seja possível compreender a proposta e o que foi alcançado;

Palavras-chave: Logo abaixo do Resumo/Abstract.

### **Capítulo 1 - Introdução**

Realmente introduz o leitor indicando quais são as direções do trabalho ? Apresenta o tema e o objeto do trabalho e contém as Referências do Estado da arte (quem está fazendo e em que nível os trabalhos da área estão hoje)?

- Justificativa/relevância do trabalho: explanação sobre porque o trabalho se justifica e quais os pontos de relevância do mesmo;
- Objetivo(s): "**somente**" o(s) objetivo(s) em uma frase. Também podem ser descritos na forma de "gerais" e/ou "específicos";
- Organização do trabalho (o que tem em cada capítulo).

Não há necessidade de reproduzir (copiar) as obras que embasam o trabalho e sim colocar o suficiente para o entendimento do trabalho e citar as referências;

### **Capítulo 2 - Embasamento Teórico ou Fundamentação Teórica**

Revisão da literatura dos tópicos que sustentam a ciência e o conhecimento, relativos ao(s) objetivo(s) e o(s) método(s) escolhido(s) para o desenvolvimento do trabalho;

### **Capítulo 3 - Material e Métodos ou Desenvolvimento do Projeto**

Descrição clara dos procedimentos e do material adotados para o desenvolvimento do trabalho (sem resultados), incluindo sua adequação ao trabalho.

Tem que responder às perguntas:

- Está com tamanho adequado (proporcional) à monografia?
- Há informação suficiente e clara sobre os materiais e sobre os métodos adotados?

Não há necessidade de reproduzir (copiar) as obras que embasam o trabalho e sim colocar o suficiente para o entendimento do trabalho e citar as referências.

### **Capítulo 4 - Resultados/Discussões**

Aqui se mostra o que o trabalho permitiu produzir, e às vezes o que pode ser comparado com outros trabalhos - aqui ficam claras se as propostas do trabalho são relevantes ou não, pois devem permitir a discussão do trabalho.

Deve responder: Os resultados estão claros em bom número (nem muito nem pouco) que permitam avaliar realmente a proposta e o que foi produzido?

### **Capítulo 5 - Conclusões**

"Fechar" com os objetivos? (respondem aos objetivos?) - aqui é que "se vende o peixe" pois irão valorizar (ou não) o trabalho realizado. Normalmente é uma parte do trabalho "um pouco desprezada", pois o autor já está "cansado...". Mas aqui é um ponto importante de medida se o trabalho tem ou não valor.

## Referências

Deve conter todas as referências **citadas no texto**. Observar as Diretrizes, pois lá estão os formatos corretos de citação.

## Apêndices

Todo o material produzido pelo autor durante o trabalho, que o mesmo julga importante disponibilizar, mas que não deve estar no corpo do trabalho, pois atrapalharia a leitura do mesmo.

## Anexos

Todo o material que não é de autoria própria, mas que é importante para completar as informações do corpo do texto (ex. datasheet).

### Outras observações **IMPORTANTES** (leia isso com atenção)

NUNCA copie texto de outro autor sem a devida forma de citação (ver em diretrizes); a cópia configura plágio! Com a Internet e/ou outras ferramentas dedicadas, é muito fácil identificar se houve cópia de texto. Se você quiser verificar a porcentagem que seu texto apresenta de similaridade com outros na internet, baixe e rode o Copy Spider, por exemplo, ou consulte outros em <http://www.escritacientifica.sc.usp.br/anti-plagio/>.

- ⇒ O tempo verbal a ser usado no texto, de forma geral, é o "PASSADO", pois o trabalho já aconteceu;
- ⇒ no texto, toda primeira vez que aparecer algum protocolo, procedimento, nome técnico, sigla, abreviatura, etc, além de explicar o que é, é necessário citar a referência. Exemplo: ...um giroscópio (referência) é um tipo de sensor...
- ⇒ figura que não é de sua autoria deve conter a fonte;
- ⇒ capriche nas figuras (uma figura bem composta quase não precisa de texto para explicá-la);
- ⇒ todas as figuras e tabelas devem ser referenciadas no texto;
- ⇒ procure manter a "Uniformidade de Notação" para o texto todo, ou seja, se denominou ou se referiu a algo ou alguém de uma certa forma, mantenha essa forma para se referir durante todo o texto;
- ⇒ não tenha medo de citar os trabalhos de outros autores (isso é imprescindível);
- ⇒ evite "muitas" referências de sites, pois são voláteis - procure boas referências nas bases consagradas como a IEEE (<http://ieeexplore.ieee.org/Xplore/home.jsp>), pois possuem artigos de ótimo nível;

- ⇒ **NÃO USE O WIKIPEDIA COMO REFERÊNCIA;**
- ⇒ todas as palavras escritas em inglês (ou em outras línguas) devem estar em itálico;
- ⇒ cuidado com o uso de "através", que significa "atravessar" algo e não por meio de ;
- ⇒ todas as obras citadas nas referências devem estar citadas no texto;
- ⇒ evite o uso de "satisfatório", "razoável" ou outra palavra que não seja precisa ou que não tenha sido definida a ordem de grandeza no texto;
- ⇒ códigos de programas devem estar em Apêndices, pois servem para comprovar o desenvolvimento e facilitar a reprodução do trabalho;
- ⇒ Anexos são informações que não são de sua autoria, mas que são importantes e que devem fazer parte da monografia para auxiliar e esclarecer o leitor.

## Apêndice B

# Apresentação do TCC

### Cuidados e orientações para a elaboração da Apresentação do TCC

Todos os meus alunos me enviam a apresentação previamente, pois faz parte do procedimento que adoto para os TCCs.

Como tem-se até 30 minutos para fazer a apresentação deve-se dimensionar a quantidade de slides para isso. Cada um tem seu "timing" com relação à quantidade de informação versus tempo disponível para apresentação. Os slides devem ser sempre muito mais visuais que textuais, ou seja, não se deve colocar frases e "ficar lendo" as mesmas. Os slides devem apresentar uma forma "clean" para que sirvam apenas de guia para a apresentação do trabalho.

Leia no site da Elétrica (/Graduação/Trabalhos de Conclusão de Curso - TCC) as DIRETRIZES GERAIS PARA ELABORAÇÃO DO TRABALHO DE FORMATURA - TCC, onde pode-se encontrar as Fichas de Avaliação que são sugeridas pelo Depto, que não são necessariamente seguidas à risca pelos avaliadores, mas que servem de bom guia para os alunos entenderem como são feitas as avaliações. Tente não utilizar fundo escuro, pois escurece o ambiente e às vezes não se consegue o visual esperado. Sempre que possível teste antes no local da apresentação.

Resumindo:

- ⇒ Prepare o seu ambiente de apresentação - mesa, cadeiras, etc., colocadas de maneira a não te atrapalhar;
- ⇒ Teste as cores que o projetor realmente projeta para que a visualização seja próxima daquela construída nos slides;
- ⇒ Evite usar fundo escuro;
- ⇒ A apresentação deve dedicar o maior tempo para o trabalho em si, suas propostas, seus resultados/discussões e conclusões;

- ⇒ Coloque um Sumário resumido da apresentação e não do trabalho todo;
- ⇒ Descarregue os slides de textos excessivos - os slides devem servir de guia para a apresentação e suporte visual para o público;
- ⇒ Slides com numeração - facilita o controle e a identificação do conteúdo;
- ⇒ Frases longas dificultam a apresentação pois induz o público à leitura e não à apresentação do palestrante;
- ⇒ Sua apresentação deve "caber" dentro de 15 a 30 minutos;
- ⇒ Não colocamos slides sobre Referências na apresentação, a menos que alguma(s) publicação(ões) seja(m) muito importante(s) a ponto de merecer destaque na apresentação;
- ⇒ O último slide deve conter "**OBRIGADO**" e não "Perguntas".
- ⇒ Como sou o orientador, eu serei o condutor de todo o ritual da defesa.



## Apêndice C

# Monografia Parcial do TCC

### Cuidados e orientações para a composição da Monografia Parcial do TCC

Trata-se de uma Monografia completa, com todas as partes de uma Monografia final.

Atente-se para as partes em **vermelho**.

- Resumo
- Introdução
- Objetivos
- Justificativas/Relevância
- Embasamento Teórico (Fundamentação Teórica-Revisão Bibliográfica)
- Material e métodos ou Desenvolvimento do Projeto
- **Resultados Preliminares**
- **Conclusões Preliminares**
- **Sequência do trabalho (indicando possíveis correções de rota do projeto)**
- **Cronograma Final (com correções se necessário)**
- Referências
- Apêndices
- Anexos

Sendo bem feito, irá poupar esforço para a redação da monografia.



## **Anexo I**

# **Aquilo que não é de sua autoria**

Texto do Anexo 1.



## **Anexo II**

**Aquilo que não é de sua autoria mas você  
julga importante colocar aqui**

Texto do Anexo 2.