

PROJETO FATEQUINO: Visão

Bruna Weber da Nóbrega – Fatec Carapicuíba

Caio César Silva Paulino – Fatec Carapicuíba

Feliphe Lorrã Serodio Jesus – Fatec Carapicuíba

Francisco Roniele Melo de Castro – Fatec Carapicuíba

Gabriel Soler Belmonte – Fatec Carapicuíba

Guilherme Carvalho Caldeira – Fatec Carapicuíba

Lucas Paes de Oliveira – Fatec Carapicuíba

Prof.(a) Mario Marques – Fatec Carapicuíba

CARAPICUIBA/SP

2020

RESUMO

Este projeto traz informações e soluções para problemas em um software na área de inteligência artificial do grupo de visão. Por ser um tema amplo para exploração e desenvolvimento, realizou-se pesquisas referente a captura de imagens por intermédio de uma câmera com o objetivo de reconhecimento de faces e movimentos. A partir daí, foi modelado um software para atender essa necessidade. Utiliza-se uma ferramenta prática para esse aprimoramento as bibliotecas para reconhecimento facial. Por não haver projetos semelhantes na Fatec Carapicuíba, esse trabalho auxiliará outros desenvolvedores para mais implementações e aplicações. Constata-se que uma implantação desse software aprimorara significativamente a interação do Fatequino.

Palavras-chave: Interação. Software. Visão.

ABSTRACT

This project brings information and solutions to problems in software in the artificial intelligence area of the vision group. As it is a broad topic for exploration and development, research was carried out regarding the capture of images through a camera with the objective of recognizing faces and movements. From there, software was modeled to meet this need. A practical tool for this improvement is used in libraries for facial recognition. As there are no similar projects at Fatec Carapicuíba, this work will assist other developers for more implementations and applications. It appears that an implementation of this software had significantly improved the interaction of Fatequino.

Keywords: Interaction. Software. Vision.

LISTA DE FIGURAS

Figura 01: Expressão de Segmentação.....	7
Figura 02: Imagem Segmentada.....	8
Figura 03: Imagem Segmentada Após Erosão.....	8
Figura 04: Exemplo Gesto (Alunos).....	11

SUMÁRIO

1	INTRODUÇÃO.....	6
2	FUNDAMENTAÇÃO TEÓRICA	6
2.1	O que é segmentação	6
2.2	Segmentação de Movimento.....	7
3	Procedimentos METODOLÓGICOS	8
3.1	Linguagem de Programação utilizada.....	8
3.2	Análise do Código	9
3.2.1	Atualização do Código.....	9
3.3	Aplicativo de Reconhecimento de Objetos.....	Erro! Indicador não definido.
4	DESENVOLVIMENTO.....	10
4.1	Grupo Anterior.....	10
4.2	Grupo Atual	10
4.3	Implementação	10
5	RESULTADOS E DISCUSSÃO	11
6	CONSIDERAÇÕES FINAIS	12
	REFERÊNCIAS	13

1 INTRODUÇÃO

Atualmente na Fatec Carapicuíba os alunos conseguem informações sobre aulas e professores somente pelo site disponibilizado (SIGA e site oficial da Fatec Carapicuíba), entretanto há alunos que possuem dificuldades para realizar consultas sobre horário de aulas e professores.

Pensando nisso foi iniciado um projeto para que este problema seja solucionado de forma rápida e eficiente. O projeto nomeado “Fatequino”, possui esse propósito de auxiliar tanto alunos que não possuem conhecimento da instituição ex: novatos, novos professores e visitantes. Auxiliando-os e facilitando em suas buscas através de uma informação simples e rápida.

O projeto está dividido em 6 grupos que realizarão o desenvolvimento de suas respectivas áreas: Controle, Interação, Mecânica 1 e 2, Visão e Web.

Neste artigo falaremos sobre a parte de visão, nosso objetivo principal é realizar reconhecimento facial de pessoas e objetos, a fim de que seja alcançado o resultado estipulado pelo Fatequino.

2 FUNDAMENTAÇÃO TEÓRICA

Utilizamos dos arquivos anteriores disponibilizados no Trello, Google Drive e GitHub para maior entendimento do que deveria ser aprimorado e para o desenvolvimento utilizamos a Internet como fonte de pesquisa.

2.1 O que é segmentação

Em visão computacional, segmentação se refere ao processo de dividir uma imagem digital em múltiplas regiões (conjunto de pixels) ou objetos, com o objetivo de simplificar e/ou mudar a representação de uma imagem para facilitar a sua análise (SALDANHA, 2009). Segmentação de imagens é tipicamente usada para localizar objetos e formas (linhas, curvas etc.) em imagens (NIELSEN, 2004).

O resultado da segmentação de imagens é um conjunto de regiões/objetos ou um conjunto de contornos extraídos da imagem (ver detecção de borda). Como resultado, cada um dos pixels em uma mesma região é similar com referência a alguma característica ou propriedade computacional, tais como cor, intensidade, textura ou continuidade. Regiões adjacentes devem possuir diferenças significativas com respeito a mesma característica(s).

Segundo Sobral (2002, p.1)

A segmentação pode seguir duas estratégias genéricas:

Descontinuidade: A partição da imagem é efetuada com base nas alterações bruscas de intensidade (ex: detecção de contornos).

Similaridade: A partição é efetuada com base na similaridade entre pixels, seguindo um determinado critério (ex: binarização, crescimento de regiões, divisão e junção de regiões).

2.2 Segmentação de Movimento

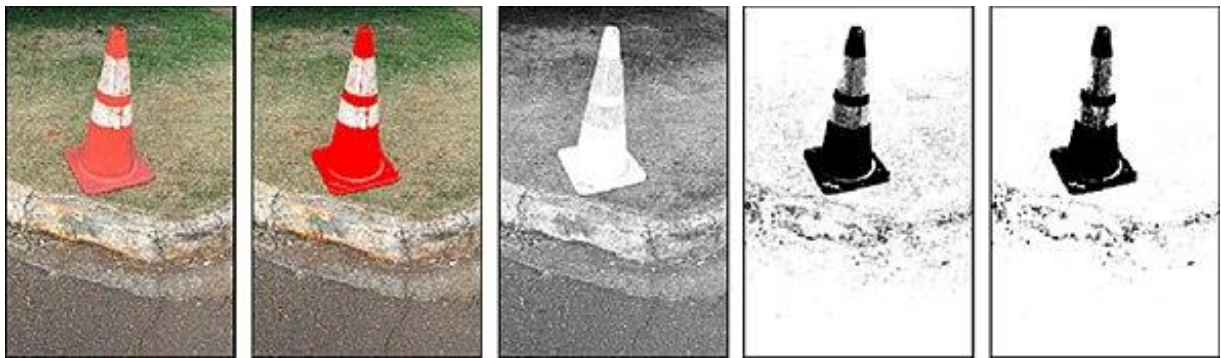
O movimento é uma poderosa pista usada por seres humanos e animais para a extração de um objeto de interesse de um fundo de detalhes irrelevantes, logo também é muito utilizada na segmentação. O modelo utilizado mais básico consiste na comparação pixel a pixel entre duas imagens, subsequentes, capturadas de um mesmo ambiente. Tal diferença pode ser expressa como mostrado na Figura 01.

$$d_{ij}(x, y) = \begin{cases} 1, & f(x, y, t_i) - f(x, y, t_i) > \theta \\ 0, & \text{caso contrario} \end{cases}$$

Figura 01: Expressão de Segmentação

Onde θ é o limiar entre os tons de cinza, isto é na imagem obtida *dij*, só existiam os pixels que apresentaram uma diferença de cor maior que θ . Com isso a imagem obtida será apenas a silhueta do objeto que se movimentou, porque o ambiente permanecerá o mesmo e será apagado com a diferença.

É possível ver um exemplo de imagem segmentada, conforme Figura 02 e Figura 03.

**Figura 02: Imagem Segmentada****Figura 03: Imagem Segmentada Após Erosão**

Porém tal método possui algumas limitações: o ambiente nas duas imagens tem que possuir uma iluminação constante, o tempo entre as duas imagens têm que ser pequeno o suficiente para pegar apenas uma silhueta do objeto e não um borrão do mesmo e grande o suficiente para ser possível a observação do movimento. Uma possível melhora para esse método é a utilização de mais imagens, onde as diferenças entre elas são acumulativas, gerando uma imagem mais nítida do objeto e diminuindo os ruídos da imagem.

3 PROCEDIMENTOS METODOLÓGICOS

3.1 Linguagem de Programação utilizada

Como nos grupos anteriores continuaremos a utilizar a linguagem Python, pois ela demonstra-se perfeita e compatível para continuarmos o projeto. Utilizaremos também o OpenCV que é uma biblioteca multiplataforma. Esta ferramenta é totalmente gratuita, voltada para estudos acadêmicos, têm a capacidade de processamento de imagens, ajudando assim, a distinguir os rostos das pessoas que serão cadastradas no Fatequino.

3.2 Análise do Código

O código disponibilizado pela equipe anterior traz informações de como o programa funciona, para melhor utilização é necessário um arquivo que recebe as imagens que são capturadas pela câmera e são tratadas (através do `ENCODE_FACE.txt` e o `RECOGNIZE_FACES_VIDEO.txt`).

ENCODE FACES: Nesse arquivo temos as bibliotecas utilizadas na aplicação, a busca de caminho das imagens, a interação com as mesmas e temos os encoding faciais salvos em um arquivo

RECOGNIZE FACES VIDEO: Já neste arquivo temos as importações de bibliotecas responsáveis pela inicialização da stream de vídeo, busca de encode facial e similares para comparação.

3.2.1 Atualização do Código

Devido a complicações com algumas bibliotecas no código utilizado pelo grupo anterior, foi recomendado e orientado pelo professor uma alteração dele. Então nos foi dado outras referências como GENT (2016) e VIANA (2018), assim implantamos outro método de reconhecimento de faces pela câmera, utilizando os arquivos `landmarks.py` e o `arquivo.dat` (que nomeamos como no projeto original). Durante o desenvolvimento usamos como base o site PyPI e o Anaconda, portanto todo o processo de instalação foi realizado dentro do ambiente virtual Anaconda.

landmarks.py: No arquivo encontram-se as bibliotecas, as coordenadas da detecção de 68 pontos de referência de faces pela câmera, que são checados com base no arquivo “.dat”.

“arquivo”.dat: Arquivo identificador de pontos de referência, que no método “shapepredictor” da biblioteca `dlib` (no arquivo `landmarks.py`), busca desse arquivo as coordenadas para a detecção dos pontos de referências.

4 DESENVOLVIMENTO

4.1 Grupo Anterior

Estava desenvolvendo o reconhecimento facial de pessoas para quando o robô entrasse em contato com a pessoa, o Fatequino pudesse reconhecê-la e auxiliá-la através de perguntas feitas pela pessoa.

4.2 Grupo Atual

Documentar os métodos de instalação no projeto, atualizar no repositório github os arquivos e se possível implementar o reconhecimento de gestos com o propósito de identificar faces e gestos. Caso alguém queira interagir com o Fatequino, o usuário deverá fazer o sinal de “V”, levantando o dedo indicador e médio de forma direta para a câmera do robô a fim de que o Fatequino tenha a ciência de sua solicitação para responder alguma dúvida, seja de alunos ou professores.

4.3 Implementação

Trabalharemos com a segmentação (processamento de imagem) para identificar formas e objetos nas imagens, com o objetivo de poder identificar possíveis rostos e gestos. Há vários tipos de segmentação, porém a que mais se adequa ao nosso projeto é a descontinuidade de movimento, pois como o Fatequino estará sempre em movimento na Fatec, os obstáculos serão móveis e não fixos (no caso de pessoas), para futuramente também poder utilizar o aceno, esse tipo de segmentação irá se adequar bem ao que foi sugerido para o projeto.

5 RESULTADOS E DISCUSSÃO

Durante o desenvolvimento do projeto encontramos algumas dificuldades que puderam ser contornadas com a alteração do código para atender as expectativas propostas ao grupo.

O código é capaz de reconhecer rostos, ao localizar a face ou gesto coloca em uma caixa na cor azul clara mostrando que está conseguindo identificar algo, em seguida faz uma marcação com contorno na cor verde com pontos na cor azul para especificar a face ou gesto, por último realiza a segmentação de imagem deixando a face ou gesto em branco com fundo preto.

Na Figura 04 é possível ver o que foi dito anteriormente e a interação do programa com o usuário imprimindo na tela a mensagem: “Olá Aluno!”.

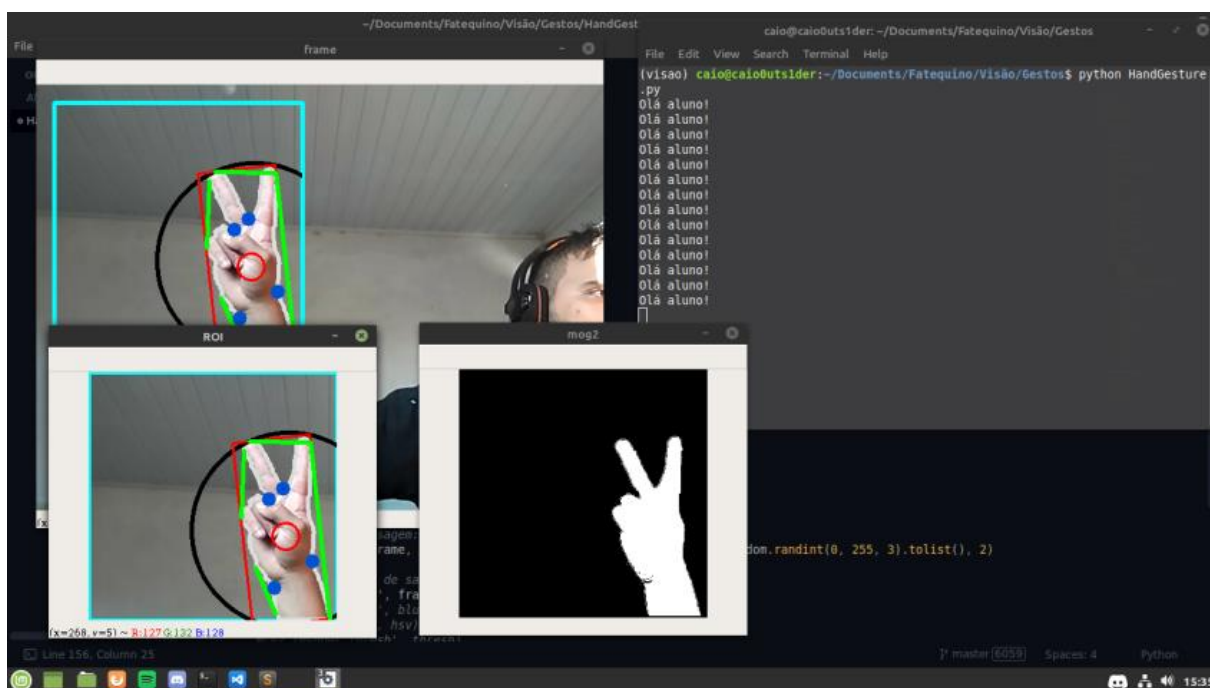


Figura 04: Exemplo Gesto (Alunos)

Algumas questões com o código anterior não puderam ser resolvidas, sendo necessário criar um com o auxílio do professor e fontes já especificadas anteriormente.

6 CONSIDERAÇÕES FINAIS

Ao desenvolver o projeto pudemos notar que a implantação desse novo código facilitará e contribuirá para a conclusão do Fatequino.

Apesar dos empecilhos durante o desenvolvimento, podemos observar que o código fará o auxílio para futuros grupos pois ele é capaz de reconhecer gestos específicos e faces realizando a segmentação de imagem. Futuramente esse código fará parte da integração no Fatequino, sendo possível interagir com o aluno ou visitante.

Todos os códigos, read-me, licenças, documentação, powerpoint e exemplos estão disponíveis com o professor responsável, no trello e github.

REFERÊNCIAS

SOBRAL, João Luís. Segmentação de Imagem. *In: Segmentação de Imagem*. [S. l.], 21 out. 2002. Disponível em:

<http://gec.di.uminho.pt/lesi/vpc0304/Aula07Segmenta%C3%A7%C3%A3o.pdf>.

Acesso em: 18 mar. 2020.

SALDANHA, Marcus S. F.; FREITAS, Dr^a Corina da Costa. Divisão de Processamento de Imagens. *In: Segmentação de Imagens Digitais: Uma Revisão*. [S. l.], 06 fev. 2009.

Disponível em:

<http://mtc-m16c.sid.inpe.br/col/sid.inpe.br/mtc-m18@80/2010/06.22.18.13/doc/106003.pdf>.

Acesso em: 18 mar. 2020.

NIELSEN, Frank; NOCK, Richard. Transactions on Pattern Analysis and Machine Intelligence. *In: IEEE Transactions on Pattern Analysis and Machine Intelligence*. [S. l.], 20 set. 2004. Disponível em: <https://ieeexplore.ieee.org/document/1335450>. Acesso em: 18 mar. 2020.

GOOGLE Cloud Vision. *In: CONHEÇA A FANTÁSTICA GOOGLE CLOUD VISION API PARA IDENTIFICAÇÃO DE OBJETOS*. [S. l.], 06 jan. 2016.

Disponível em: <https://mundoapi.com.br/materias/conheca-a-fantastica-google-cloud-vision-api-para-identificacao-de-objetos/>. Acesso em: 18 mar. 2020.

JR, Valdir Stumm; DORNELES, Elias. ACESSANDO APIS REST COM PYTHON. *In: ACESSANDO APIS REST COM PYTHON*. [S. l.], 06 jan. 2016.

Disponível em: <https://pythonhelp.wordpress.com/2014/07/25/acessando-apis-rest-com-python/>. Acesso em: 20 maio 2020.

OPENCV. *In: OPENCV*. [S. l.], 10 fev. 2014.

Disponível em: <https://pt.wikipedia.org/wiki/OpenCV>. Acesso em: 20 maio 2020.

GENT, Paul Van. *In: EMOTION RECOGNITION USING FACIAL LANDMARKS, PYTHON, DLIB AND OPENCV*. [S. l.], 05 ago. 2016. Disponível em:

<http://www.paulvangent.com/2016/08/05/emotion-recognition-using-facial-landmarks/>.

Acesso em: 06 jun. 2020.

VIANA, Suzana. *In: CONFIGURANDO O AMBIENTE DLIB + PYTHON: GUIA PARA INICIANTES*. [S. l.], 13 jul. 2018. Disponível em:

<https://medium.com/@suzana.svm/configurando-o-ambiente-dlib-python-guia-para-iniciantes-81cdcffc937e>. Acesso em: 06 de jun. 2020.

ANACONDA - Anaconda Inc. *In: Anaconda Individual Edition*. Disponível em:

<https://www.anaconda.com/products/individual>. Acesso em: 20 de jun. 2020.

PYPI - Python Software Foundation. *In: The Python Package Index*. Disponível em:

<https://pypi.org/>. Acesso em: 20 de jun. 2020.

“O conteúdo expresso no trabalho é de inteira responsabilidade do(s) autor(es).”