

UFMA
CIÊNCIA DA COMPUTAÇÃO
VISÃO COMPUTACIONAL

Isabel Souza de Carvalho
Luiza Helena Vieira

FussyFood:

An Application to
Navigate Food Allergies
& Dietary Restrictions

Sameep Bagadia
Rohit Mundra

Sumário

- Introdução
- Funcionamento do Aplicativo
- Pré-processamento da imagem
- Reconhecimento ótico de caracteres e busca no banco de dados
- Análise de erros e resultados
- Trabalhos futuros
- Referências

Introdução

- 43-72 milhões de pessoas alérgicas a amendoim
- 11 milhões de pessoas vegetarianas (América)
- feedback rápido ao usuário

Funcionamento do aplicativo

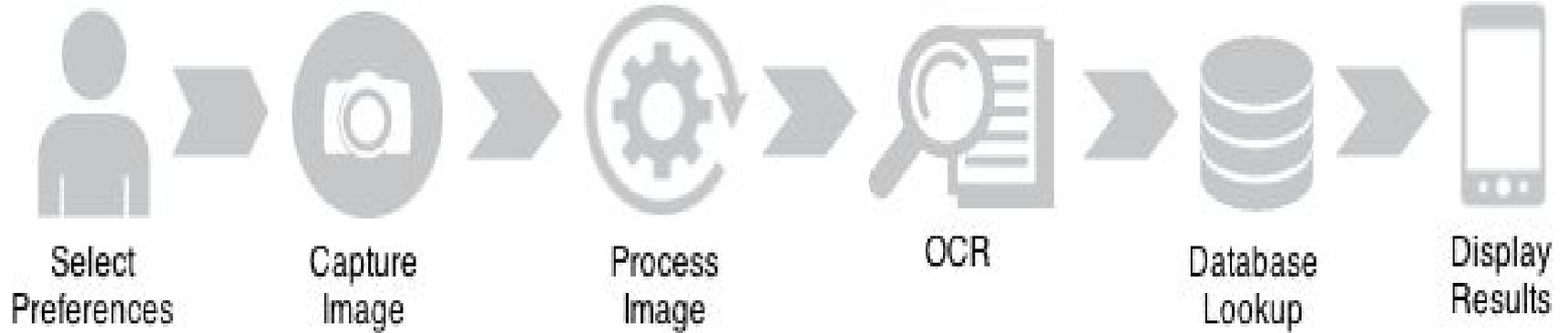


Figura 1: Fluxo de processamento

Funcionamento do aplicativo

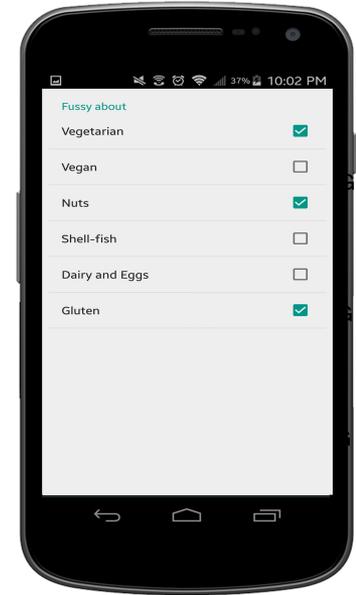
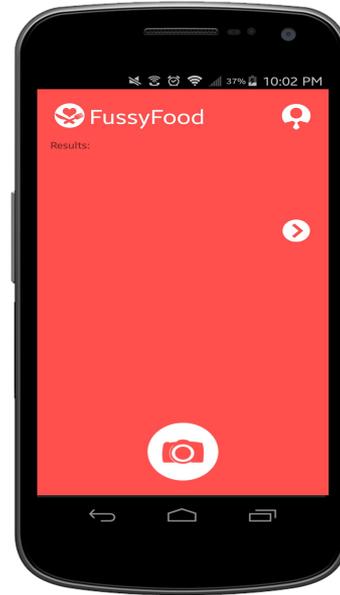


Figura 2: (a) Tela Inicial

(b) Painel de preferências

Funcionamento do aplicativo



Capture
Image

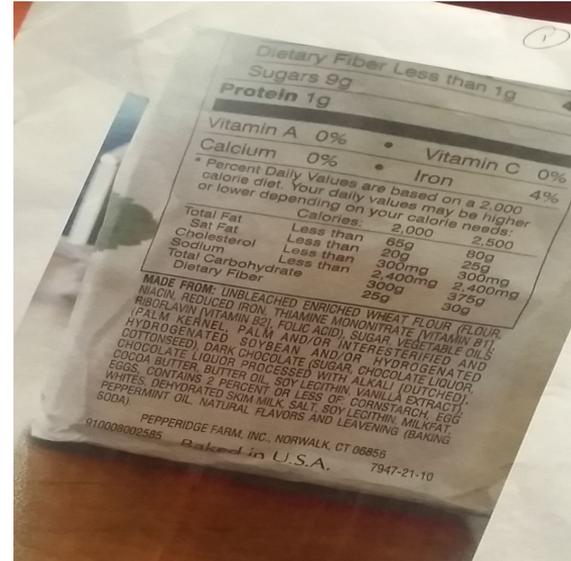
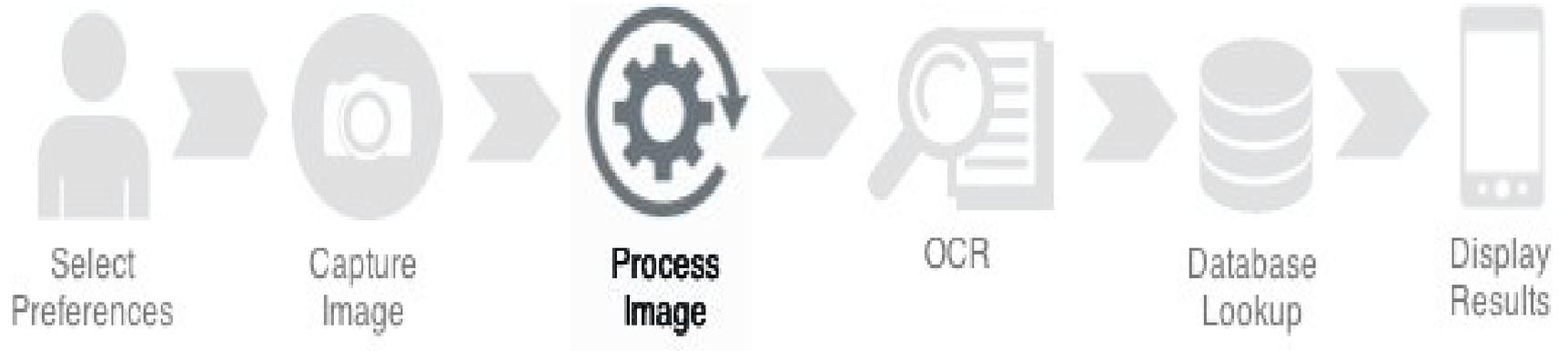


Figura 3: Imagem Original

Funcionamento do aplicativo



Pré-processamento da Imagem

- Locally Adaptive Thresholding
 - RGB > escalas de cinza
 - Para evitar Threshold incorreto > Limiarização Adaptativa Local
 - Método de Otsu > blocos onde a variância é elevada

Observações:

- A variância, σ_i^2 , da imagem digital i representa o valor de desvio dos níveis de cinza da imagem em relação ao nível de cinza médio.
- O método de limiarização bimodal de Otsu é baseado na análise de discriminante. A operação de limiarização é considerada como sendo o particionamento dos pixels de uma imagem de I níveis de cinza em duas classes, C_0 e C_1 , que podem representam o objeto e o fundo, ou vice-versa. [1]

Pré-processamento da Imagem

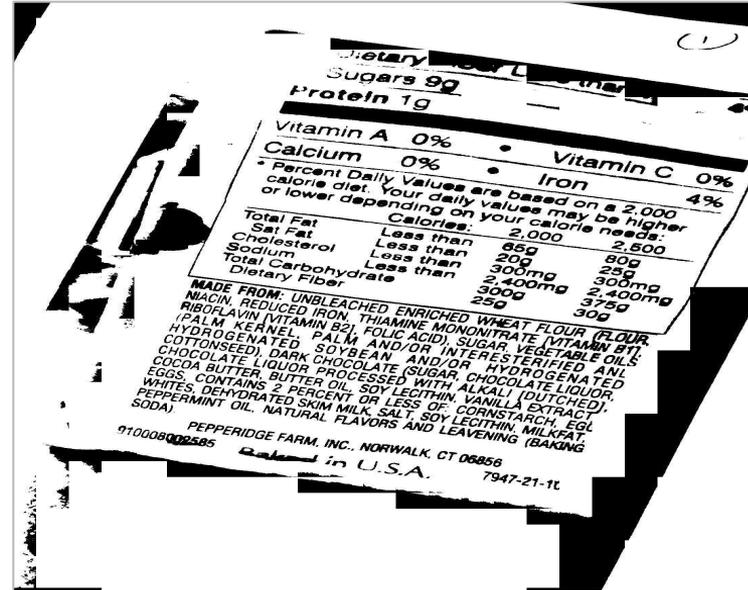
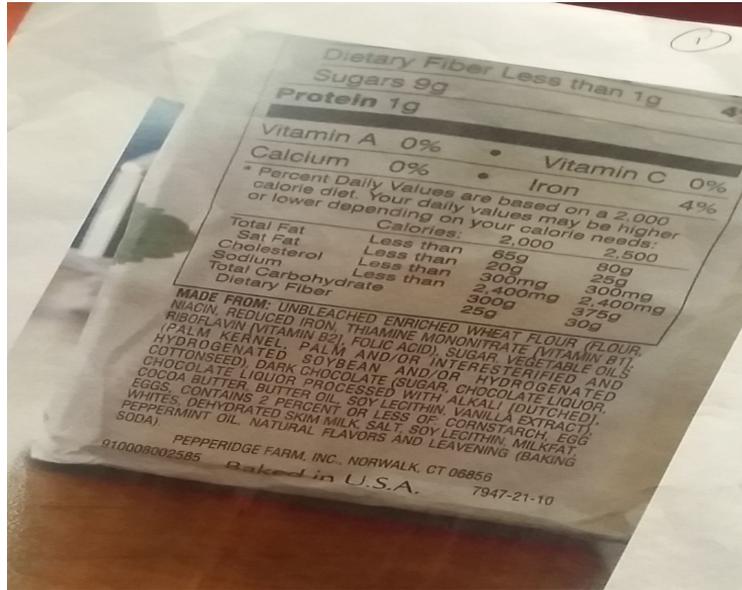


Fig. 4: Locally Adaptive Thresholding

Pré-processamento da Imagem

- Probabilistic Hough Transform for Auto-rotation
 - Imagens inclinadas não serão processadas corretamente
 - Transformada de Hough (HT) > detecção de formas que são facilmente parametrizadas (linhas, círculos, elipses, etc.)

Observações:

- Transformada de Hough

O método envolve a transformação do plano de coordenadas da imagem para um espaço parametrizado. Considerando a aproximação de retas, a equação da linha pode ser expressa como:

$$r = x \cdot \cos\theta + y \cdot \sin\theta$$

r é a distância do ponto à origem

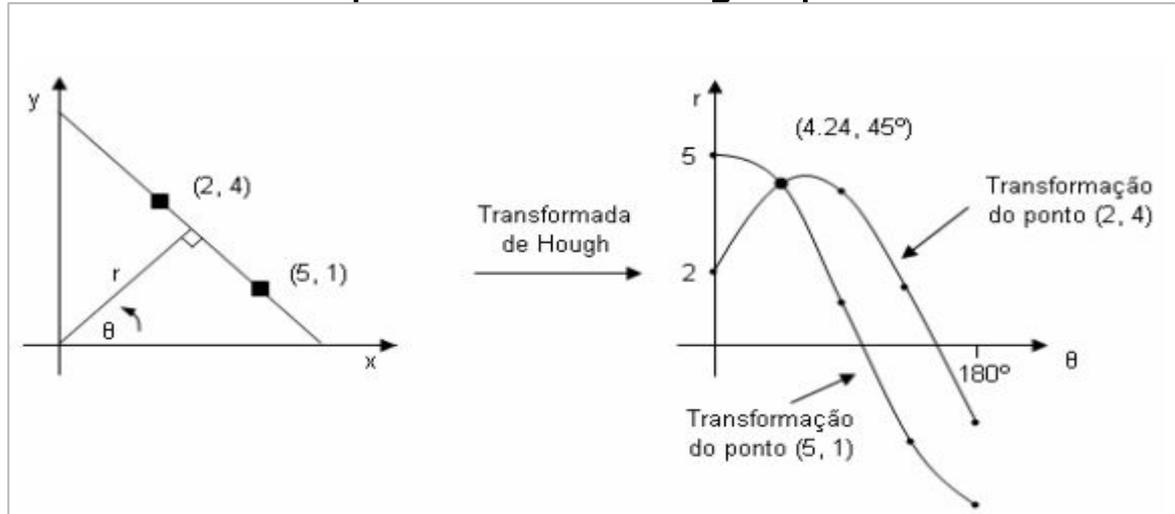
θ é o ângulo formado entre r e o eixo axial

o par (r, θ) define espaço de Hough

Observações:

- Transformada de Hough

Cada pixel preto da imagem de coordenada (x, y) é mapeada com uma nova coordenada (r, θ) no plano de Hough para todos os valores possíveis de θ .



Pré-processamento da Imagem

- Probabilistic Hough Transform for Auto-rotation
 - $r = 1$ pixel
 - $\theta = 1$ grau
 - Segmentos de linha $< \frac{1}{3}$ da largura da imagem
 - No máximo 20 pixels entre os pontos a serem detectados como parte do mesmo segmento de linha.

Pré-processamento da Imagem

- Probabilistic Hough Transform for Auto-rotation
 - Removidas linhas com ângulo > 45 graus
 - Mediana dos ângulos dos segmentos de linha \longrightarrow ângulo de inclinação do imagem
 - Ângulo de inclinação \longrightarrow Rotaciona a imagem

Pré-processamento da Imagem

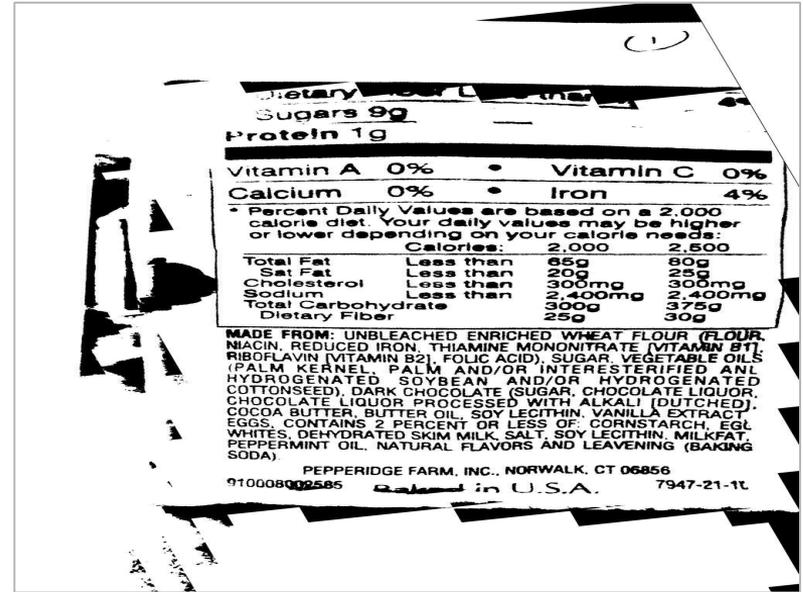
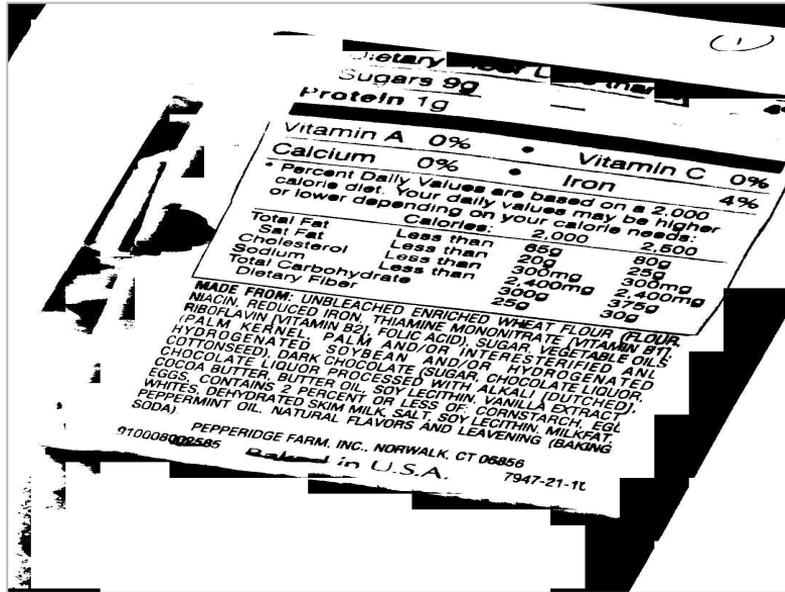


Fig. 5: Auto-rotation

Pré-processamento da Imagem

- Remoção do background
 - Método de detecção de bordas Canny: eliminação de ruído
 - Aplicação de filtro: para destacar os pixels brancos nas regiões de bordas detectadas
 - Remoção dos ruídos: threshold + equalização

Pré-processamento da Imagem

- Remoção do background



Figura 6: Utilizando Detecção de Bordas de Canny

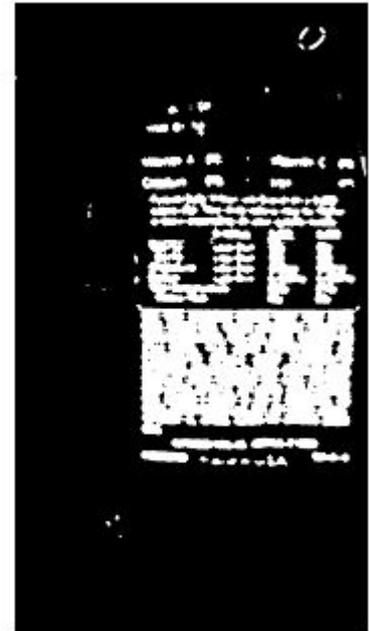


Figura 7: Remoção de Ruídos no Background

Pré-processamento da Imagem

- Detecção da Bounding Box
 - Nem todo o texto é relevante
 - Melhor processamento da imagem
 - Dilatar a imagem até encontrar a menor área com mais texto: área da lista de ingredientes
 - F-score: utilizado para maximizar os pixels brancos sem comprometer a imagem
 - Precisão: região de imagem não contida na bounding box
 - Recall: fração de pixels brancos contidos na bounding box

Pré-processamento da Imagem

- Detecção da Bounding Box



Figura 8: F-score maximization

$$\text{maximize } (1 + \beta^2) \frac{\text{precision} \cdot \text{recall}}{\beta^2 \text{precision} + \text{recall}}$$

F-score: utilizado para detectar a bounding box

Figura 9: Bounding box



Reconhecimento óptico de caracteres e busca no banco de dados

- Aplica threshold e envia para Tesseract OCR engine: retorna uma string de texto
 - Tesseract: biblioteca open-source para Android
- Divide a string em caracteres para encontrar palavras
- Procura no banco de dados e compara com as preferências do usuário

Reconhecimento óptico de caracteres e busca no banco de dados



MADE FROM: UNBLEACHED ENRICHED WHEAT FLOUR (FLOUR, NIACIN, REDUCED IRON, THIAMINE MONONITRATE (VITAMIN B1), RIBOFLAVIN (VITAMIN B2), FOLIC ACID), SUGAR, VEGETABLE OILS (PALM KERNEL, PALM AND/OR INTERESTERIFIED AND HYDROGENATED SOYBEAN AND/OR HYDROGENATED COTTONSEED), DARK CHOCOLATE (SUGAR, CHOCOLATE LIQUOR, CHOCOLATE LIQUOR PROCESSED WITH ALKALI (DUTCHED), COCOA BUTTER, BUTTER OIL, SOY LECITHIN, VANILLA EXTRACT, EGGS, CONTAINS 2 PERCENT OR LESS OF: CORNSTARCH, EGG WHITES, DEHYDRATED SKIM MILK, SALT, SOY LECITHIN, MILKFAT, PEPPERMINT OIL, NATURAL FLAVORS AND LEAVENING (BAKING SODA)

Figura 10: Figura utilizada para processamento

Análise de Erros e Resultados

- Métricas de avaliação: as duas primeiras garantem que ingredientes listados não sejam perdidos, gerando falso alerta, e a terceira garante que não-ingredientes sejam detectados

Análise de Erros e Resultados

Metric	Scenario 1	Scenario 2	Scenario 3
Metric 1	0	0.760	0.742
Metric 2	1	0.240	0.258
Metric 3	∞	2.079	0.0

Tabela 1: Análise de Erros

Métrica 1: Fração de palavras detectadas como ingredientes e são ingredientes

Métrica 2: Fração de palavras não detectadas como ingredientes e são ingredientes

Métrica 3: Fração de palavras detectadas como ingredientes e não são ingredientes

Cenário 1: Imagens em escala de cinza

Cenário 2: Imagens em escala de cinza e auto-rotacionadas

Cenário 3: Imagens em escala de cinza, auto-rotacionadas e recortadas

Análise de Erros e Resultados

- Cenário 1: resultados indesejáveis
- Cenário 2: apesar de melhor, ainda detecta não-ingredientes como ingredientes
- Cenário 3: ótima performance e capturar ingredientes, mas não tão boa em evitar não-ingredientes

Trabalhos Futuros

- Integrar um corretor: erros causados pelo não reconhecimento de alguns caracteres
- Compartilhar um relatório sobre a comida: visa aumentar a usabilidade do produto e relacionar mercados que se importam com saúde no geral e não tem nenhuma restrição
- Processar imagens em embalagens de outros formatos (ex: embalagens cilíndricas)

Referências

W. Bieniecki, S. Grabowski, and W. Rozenberg, “Image preprocessing for improving ocr accuracy,” in *Perspective Technologies and Methods in MEMS Design, 2007. MEMSTECH 2007. International Conference on*, pp. 75–80, IEEE, 2007.

T. E. de Campos, B. R. Babu, and M. Varma, “Character recognition in natural images,” in *Proceedings of the International Conference on Computer Vision Theory and Applications, Lisbon, Portugal, February 2009*.

R. Smith, “An overview of the tesseract ocr engine,” in *Proceedings of the Ninth International Conference on Document Analysis and Recognition - Volume 02, ICDAR '07, (Washington, DC, USA)*, pp. 629–633, IEEE Computer Society, 2007.

R. Casey and E. Lecolinet, “A survey of methods and strategies in character segmentation,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 18, pp. 690–706, Jul 1996.