

Extração de dados de tabela nutricional com reconhecimento ótico de caracteres

L. Brito

Resumo — Considerando que desde 2001 tornou-se efetiva a Resolução da Diretoria Colegiada (RDC) que obriga a rotulagem de produtos alimentícios e bebidas, os fabricantes começaram a disponibilizar os valores dos alimentos fabricados, todavia grande parte desses dados ainda têm um acesso dificultado, em que em sua maioria são disponibilizados em forma de imagens. Dessa forma o objetivo deste trabalho foi desenvolver uma *pipeline* que aumentasse a acurácia de reconhecimento ótico de caracteres (do inglês, Optical Character Recognition (OCR)) juntamente à linguagem de programação Python, a biblioteca OpenCV e o algoritmo *k-means* (K-médias). Os resultados com a imagem no final da *pipeline*, chegaram a uma diferença de 1,4 e 0,6 de médias de caracteres e palavras, respectivamente, em relação a um reconhecimento feito por um ser humano alfabetizado.

Palavras-chaves— OCR, tabela nutricional, tesseract OCR, visão computacional

I. INTRODUÇÃO

A rotulagem nutricional de alimentos industrializados se tornou obrigatória nos produtos brasileiros no ano de 2001^{[3][4]}, por meio da RDC nº 40, de 21 de março de 2001, que determina o Regulamento Técnico para a Rotulagem Nutricional Obrigatória de Alimentos e Bebidas Embalados e por meio da RDC nº 39, de 21 de março de 2001, que apresenta a Tabela de Valores de Referência para Porções de Alimentos e de Bebidas Embalados para Fins de Rotulagem Nutricional.

Desse modo a tabela sugere um apoio à população para a escolha dos alimentos, conscientização com informações nutricionais dos alimentos e no combate de doenças comuns como a obesidade, diabetes e doenças crônicas degenerativas associadas aos hábitos alimentares da população^[18], sendo útil para a construção de cardápios balanceados e dietas saudáveis.

Embora as informações nutricionais sejam disponibilizadas na embalagem de todos os produtos alimentícios brasileiros, *internet* e em alguns sites de fabricantes, a quantidade de pessoas que se interessa por essa informação normalmente é baixa, todavia os setores de pesquisas em alimentos vem crescendo e os estudos relacionados a essa área também^[1], porém na maioria das vezes os dados das tabelas nutricionais são encontrados em

formato de imagem, dificultando o trabalho com os dados, visto que é necessária uma transcrição manual, possibilitando assim, um salvamento em banco de dados, execução de modelos de processamento de linguagem natural, do inglês, *Natural Language Processing* (NLP) e semelhantes.

Como alternativa para resolver esse problema, as pesquisas de Shepard em 1977, já buscavam criar um meio de reconhecimento de caracteres^[5] e hodiernamente esse assunto ainda possui grande relevância e aprimoramentos. O sistema de OCR mais conhecido e que é considerado estado da arte atualmente é a *engine open source* Tesseract OCR^[11], a qual foi desenvolvido por Ray Smith na HP Labs Bristol por 10 anos^[20] e foi enviado ao *The Fourth Annual Test of OCR Accuracy*^[17], contando com uma *pipeline* de processos para que a *engine* reconheça os caracteres^[11]. Todos os códigos do tesseract encontram-se disponíveis em <https://github.com/tesseract-ocr/tesseract>.

Em conjunto com essa ferramenta destaca-se o uso de métodos de análise de imagens com visão computacional e aprendizado de máquina para que os resultados sejam melhores e que o trabalho de reconhecimento seja facilitado.

Dessa forma, o objetivo do presente trabalho é realizar a extração de dados de tabelas nutricionais na *internet*, com o uso de tecnologias de OCR, de forma que apresente tratativas de visão computacional e aprendizado de máquina para facilitar a extração desses dados.

II. MATERIAIS E MÉTODOS

Como pré-requisito para que houvesse a tratativa para melhoria de desempenho quanto à qualidade para a seleção com o Tesseract OCR, foi utilizada a linguagem Python com as bibliotecas OpenCV e Scikit Learn.

A. Tesseract OCR

A *engine* Tesseract OCR teve seu início motivada como um projeto de PhD, em que não obtinha boa acurácia em seus resultados, todavia na HP Labs Bristol foi trabalhada e daí começou a apresentar bons resultados, onde foi submetida ao campeonato de OCRs (*The Fourth Annual Test of OCR Accuracy*) e desde então tem apresentado bons resultados, a ponto de tornar-se o estado da arte em OCR, e repercussão, quando em 2005 foi liberada de modo *open source*.

Para realizar o reconhecimento das palavras, a *engine* conta com uma *pipeline step-by-step* que contempla desde encontrar as linhas, até o reconhecimento dos caracteres com base no seu contorno^[20], utilizando de um modelo de *Long short-term memory* (LSTM) em sua versão 4.0^[16].

Além disso, vale destacar que a própria documentação da *engine* Tesseract conta com sugestões para a melhoria de qualidade, aplicando filtros, realizando redimensionamento, binarização, remoção de ruídos, dilatação e erosão, rotação e afins^[8].

B. Imagens de tabela nutricional

As imagens utilizadas para a aplicação de OCR, assim como os filtros que propusessem uma melhoria na qualidade do reconhecimento, foram coletadas em sites de alimentos, visto que obrigatoriamente os fabricantes precisam disponibilizar a tabela nutricional^{[3][4]}. Todavia, para que não houvesse indicação ou implicações com as marcas, as imagens que continham o nome ou alguma informação específica da marca foram discretizadas, sobrepondo essa informação pela cor de fundo da tabela.

Pelas imagens serem da *internet*, não foi estabelecido um padrão de tamanho em largura e altura na coleta, assim como também foi testado com uma variedade de cores.

C. OpenCV

Open Computer Vision Library, mais conhecido como OpenCV, é uma biblioteca *open source* que tem como objetivo moldar uma infraestrutura de fácil acesso e com ótima estrutura de processamento, possibilitando com mais de quinhentas funções^[2], o trabalho com imagens, em diversas subáreas, em que pode-se citar, aprendizado de máquina, imagens médicas, calibração de câmeras, entre outras^[2]. Seu código fonte foi escrito em C e C++, todavia contempla de *wrappers*, dentre várias linguagens, em Python, a qual foi utilizada no presente trabalho.

D. Definição de acurácia

Para a acurácia dos testes realizados, assim como para a classificação de desempenho do algoritmo e escolha da melhor abordagem, utilizou-se da menor diferença de palavras e caracteres reconhecidos entre todas as abordagens testadas, considerando que o reconhecimento considerado correto foi realizado por um humano alfabetizado.

Todavia, dentre os métodos de avaliação destaca-se a fórmula empregada no *The Fourth Annual Test of OCR Accuracy*^[17], a qual pode ser representada por:

$$A = \frac{xc - ec}{xc} \quad (1)$$

Em que tem-se A representando a acurácia, xc para simbolizar a quantidade de caracteres presentes na frase (tabela nutricional, no caso deste trabalho) e por fim, o ec simbolizando o número de caracteres errados após a aplicação dos tratamentos e que são fruto do reconhecimento ótico.

Vale citar que esporadicamente após a aplicação do OCR, há divergência entre a quantidade de caracteres reconhecidos e a quantidade de caracteres que realmente estão presentes na imagem, em que ocorre uma geração de caracteres estranhas ou falta de detecção em demasia^[17]. Dessa forma caso o algoritmo venha a comportar-se de forma diferente da padrão, esses comportamentos serão considerados falhas.

E. Pipeline utilizada

Tendo em vista que o Tesseract OCR já conta com uma sequência de procedimentos para o reconhecimento de caracteres, no presente trabalho também foi criada uma *pipeline* para facilitar o reconhecimento de caracteres (Fig. 1), algo muito comum para trabalhos de visão computacional, como o trabalho realizado por Harraj em 2015^[9].

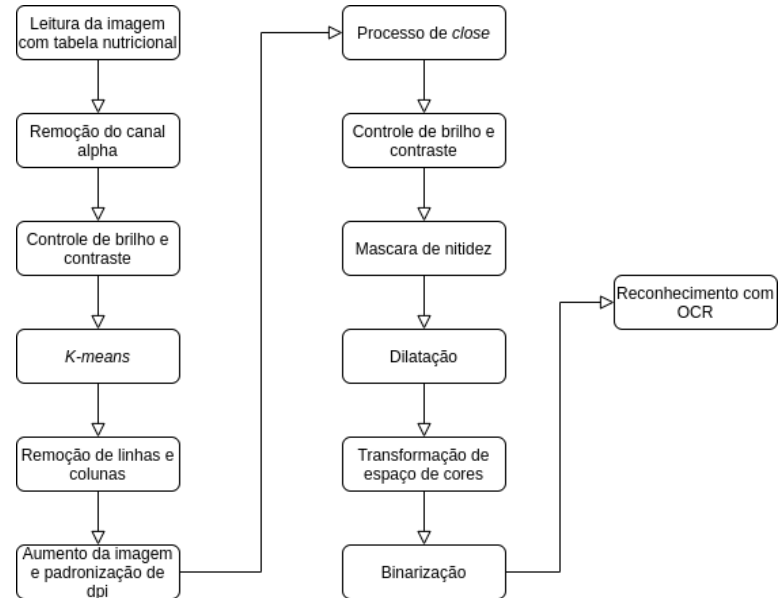


Fig. 1. Pipeline para utilizada para o reconhecimento de caracteres com Tesseract OCR. Fonte: do autor.

a. Remoção do canal alfa

Seguindo o documento de que apresenta uma série de filtros e transformações na imagem, que provê melhorias quanto à qualidade de processamento realizado pelo Tesseract OCR^[14], um dos passos está ligado na remoção do canal alpha, que encontra-se presente nas imagens de formato *Portable Network Graphics* (PNG), com o propósito de definir a opacidade dos pixels^[12].

Dessa forma, o pixel (px.) passa a ter apenas três camadas de cores, sendo eles nas cores vermelho, verde e azul (Fig. 2), as quais possuem um valor entre 0 e 255.

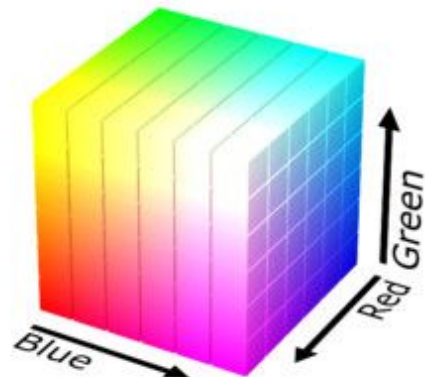


Fig. 2. Representação do espaço de cor RGB. Disponível em: https://docs.opencv.org/3.4/da/d97/tutorial_threshold_inRange.html. Acesso em: 22 de julho de 2020. Fonte: Wikimedia Commons

b. Controle de brilho e contraste

O controle de brilho e contraste também é comum para o trabalho com imagens, sendo que já foram propostas várias abordagens novas de melhoria para essas técnicas^{[9][21][7][10][19]}.

Todavia a fórmula proposta para a realização de controle de brilho e contraste, foi a seguinte:

$$g(i,j) = \alpha \cdot f(i,j) + \beta \quad (2)$$

Em que α é o valor determinado para o controle de contraste e β é o valor que utiliza-se para controle do brilho. Destacando que $\alpha > 0$ e normalmente está no intervalo de 1 à 100.

$g(i,j)$ indica o resultado da equação.

$f(i,j)$ é responsável por representar a entrada, sendo que i representa a posição horizontal do pixel/valor e j simboliza a posição vertical do pixel/valor.

Essa mesma fórmula e abordagem foi utilizada por Harraj em 2015 e é a fórmula recomendada na documentação oficial do OpenCV (https://docs.opencv.org/3.4/d3/dc1/tutorial_basic_linear_transform.html).

c. K-means

O *K-means* é um método não supervisionado que realiza agrupamento de dados de um *dataset* de acordo com as suas similaridades e em quantidades desejadas (k). Ou seja, o algoritmo gera um ponto/dado inicial randômico como centroide para um determinado *cluster* e depois tenta atribuir os demais dados do dataset aos *clusters* de acordo com a sua proximidade. Para isso, o algoritmo usa técnicas de distância euclidiana^[13].

Vale ressaltar que os pontos centroides são atualizados conforme as classificações que são feitas no algoritmo, e os mesmos param de atualizar quando o algoritmo cumpre seu objetivo, há poucas interações dos dados com diferentes *clusters* ou através de um número fixo de iterações^[6].

d. Remoção de linhas e colunas

O processo de remoção de linhas conta com uma *pipeline* própria (Fig. 3).

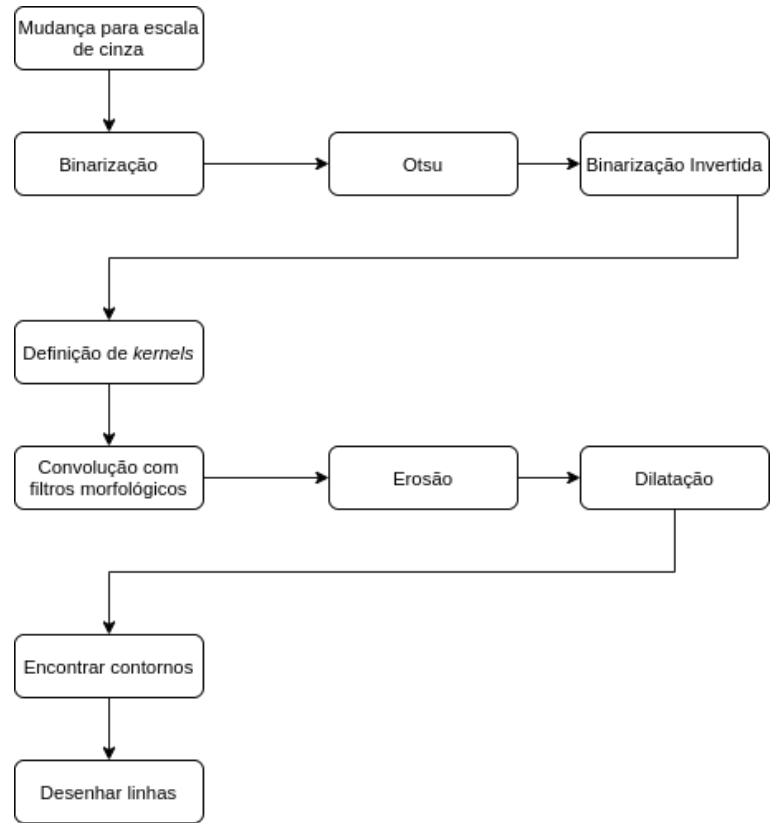


Fig. 3. Processos realizados para a remoção de linhas e colunas das tabelas. Fonte: do autor.

Esse método de remoção de linhas e colunas trabalha como um eliminador de ruídos, de sorte que haja destaque na região de interesse, do inglês *Region Of Interest* (ROI), ao remover, da imagem, características que possam vir a atrapalhar no método de OCR.

O processo de binarização (II.E.j) inicia a pipeline contando com uma definição de *kernels* horizontal e vertical, respectivamente, com uma matriz de 1×25 e 25×1 , para a realização da convolução com os filtros morfológicos de *open* da biblioteca OpenCV^[23] (Fig. 4), o qual contempla um processo de erosão, seguido de um processo de dilatação (II.E.h) sobre a imagem binarizada, de modo que encontre as retas que desenharam as linhas e colunas da tabela nutricional e sobre os mesmos, desenha linhas da mesma cor do fundo da imagem.



Fig. 4. Representação do processo de *open*. Disponível em: https://docs.opencv.org/trunk/d9/d61/tutorial_py_morphological_ops.html.

Acesso em: 22 de julho de 2020.

e. Aumento da imagem e padronização de dpi

Por padrão a documentação de melhorias do Tesseract OCR conta com uma necessidade mínima de *Dots Per Inch* (DPI) para realizar um bom reconhecimento. Esse valor é parametrizado em 300 DPIs, que é um valor muito comum para digitalização.

Além disso, a biblioteca OpenCV conta com uma função de reshape em que é possível passar o valor de altura e largura desejado para a imagem.

f. Processo de *close*

O processo de *close* da biblioteca OpenCV enquadra-se na sessão de filtros morfológicos, em que o *close* conta com dois *kernels*^[23] que realizam, respectivamente, um processo de dilatação e erosão, sendo eficiente para a remoção de falhas entre os pixels (Fig. 5).



Fig. 5. Representação do processo de *close*. Disponível em: https://docs.opencv.org/trunk/d9/d61/tutorial_py_morphological_ops.html. Acesso em: 22 de julho de 2020.

g. Mascara de nitidez

Para realizar um destaque quanto às linhas da imagem, comumente realiza-se um filtro de nitidez com uma fórmula Gaussiana^[15]. Dessa forma o filtro de máscara de nitidez matematicamente pode ser representada por:

$$dst(x, y) = src(x, y) - src_{blur}(x, y) \quad (3)$$

Sendo assim $dst(x, y)$ é o resultado, em que $src(x, y)$ é a entrada e $src_{blur}(x, y)$ é a entrada modificada por um filtro Gaussiano composto por um *kernel* e um valor de sigma, assim, removendo as bordas embasadas da imagem original e dando mais nitidez às imagens.

h. Dilatação

O processo de dilatação conta com um aumento dos pixels em relação ao valor máximo dos pixels vizinhos. Sua fórmula pode ser definida como:

$$dst(x, y) = \max_{(x', y'): \text{element}(x', y') \neq 0} src(x + x', y + y') \quad (4)$$

Em que $dst(x, y)$ é a saída, $\max_{(x', y'): \text{element}(x', y') \neq 0}$ é o valor máximo dentre os vizinhos das coordenadas de x e y , e por fim, $src(x + x', y + y')$ são as entradas.

i. Transformação de espaço de cores

As imagens coletadas encontram-se no espaço de cor RGB, todavia, durante o processo, as imagens foram transformadas para o espaço de cinza (*grayscale*), além de que a imagem em escala de cinza é um pré requisito para a transformação binária.

Como supracitado, na seção II.E.a, as imagens possuem três canais/camadas, sendo RGB e quando há uma transformação para a escala de cinza, essas três camadas se homogeneizam. Para que esse processo de homogeneização aconteça é necessário utilizar a seguinte fórmula (4), a qual também foi utilizada por Harraj em 2015 e encontra-se como a fórmula padrão para a transformação de RGB para escala de cinza, estando implementada na biblioteca OpenCV.

$$Y = 0,299 \cdot R + 0,587 \cdot G + 0,114 \cdot B \quad (5)$$

Sendo que Y representa a saída, com o valor do pixel na escala de cinza. O R simboliza o valor do pixel na camada vermelha, assim como o G e o B representam valor do pixel em suas respectivas camadas, verde e azul.

j. Binarização

A binarização conta com a definição de um limiar para a imagem, de modo que a mesma necessita ser composta por apenas uma camada. Assim, ocorre uma convolução em que se o valor do pixel for maior que o limiar determinado anteriormente, ele recebe um valor fixo, caso contrário, receberá outro.

Vale destacar que esses valores por padrão são 0 e 1 e sua fórmula pode ser representada por:

$$dst(x, y) = \begin{cases} valorMaximo & \text{se } src(x, y) > T(x, y) \\ 0 & \text{senao} \end{cases} \quad (6)$$

Sendo que $dst(x, y)$ representa a saída com o mesmo tamanho que a entrada $src(x, y)$, tendo em vista que o resultado de cada posição recebe o *valorMaximo* quando possui uma entrada maior que o limiar $T(x, y)$.

Juntamente a técnica de binarização é comum utilizar da técnica de Otsu, que encontra um valor limiar global com menor variância dentro de determinadas classes^[14], dessa forma, o valor de *threshold* não precisa ser escolhido, pois, o mesmo é calculado automaticamente^[22].

III. RESULTADOS E DISCUSSÕES

A. Amostragem

Para a realização dos testes, foi feita uma coleta aleatória de 10 tabelas nutricionais na *internet*, em que apenas uma delas continha informação que era específica do produto. Dessa forma a tabela com informações específicas passou por um pré tratamento, em que foi removida essa informação.

Vale destacar que todas as imagens de tabela estavam no formato de PNG.

As tabelas podem ser acessadas a partir do seguinte link: https://github.com/Lucs1590/ocr_tests/tree/master/tables.

B. Controle de brilho e contraste

Quanto ao controle de brilho e contraste, utilizou-se da equação 2 para que foram obtidos bons resultados, sendo que nas duas vezes que esse método é utilizado durante os testes, o padrão foi o mesmo, ou seja, com o alfa (α) = 1 e o beta (β) = 0,5.

Em um primeiro momento, a configuração utilizada, baseou-se em Harraj, com um alfa de 1,4 e um beta de 0,5, todavia ao ter uma elevação maior de alfa, percebeu-se que o resultado das imagens estavam demasiadamente claros, o que poderia omitir os caracteres desejados, dessa forma foi feito uma mudança quanto aos parâmetros.

Informações Nutricionais	Qtde.por porção*	%VD**
Valor energético	147kcal=605kJ	7%
Carboidratos	19 g	6%
Proteínas	2,0 g	3%
Gorduras totais	7,2 g	13%
Gorduras saturadas	3,7 g	17%
Gorduras trans	zero g	**
Gorduras Monoinsaturadas	2,1 g	**
Gorduras Poliinsaturadas	0,5 g	**
Colágeno	zero mg	0%
Fibra alimentar	1,1 g	4%
Sódio	90 mg	4%

*Valores Diários com base em uma dieta de 2.000 kcal ou 8.400 kJ. Seus valores diários podem ser maiores ou menores dependendo de suas necessidades energéticas.
** (%)/VD não estabelecido.

Fig. 6. À esquerda imagem com as configurações definitivas. À direita imagem com as configurações de Harraj. Fonte: do autor.

C. Remoção de linhas e colunas com auxílio de *K-means*

Após a otimização da imagem, foi realizada a remoção de linhas e colunas ao desenhar uma reta com 2px em cima dos traços que formavam as linhas da tabela e suas colunas.

O método foi muito efetivo em tabelas com o fundo branco, visto que a linha que era desenhada com 2px. também era da cor branca (Fig. 7), todavia, percebeu-se que a cor não podia ser definida, visto que os resultados em tabelas coloridas não eram ocultações das linhas e colunas ao homogeneizar com a mesma cor de fundo, mas a formação de novos traços na cor branca, como se observa na Figura 8.

Quantidade por porção Porção de 30 g (1 fatia)		%VD (*)
Valor Energético	101 kcal = 423 kj	5
Carboidratos	0,4 g	0
Proteínas	7,1 g	9
Gorduras Totais	7,9 g	14
Gorduras Saturadas	5,0 g	23
Gorduras Trans	0,3 g	**
Fibra Alimentar	0	0
Sódio	263 mg	11
Cálcio	228 mg	23

* % Valores Diários de Referência com base em uma dieta de 2.000 kcal ou 8.400 kJ. Seus valores diários podem ser maiores ou menores dependendo de suas necessidades energéticas.

** VD não estabelecido.

Quantidade por porção Porção de 30 g (1 fatia)		%VD (*)
Valor Energético	101 kcal = 423 kj	5
Carboidratos	0,4 g	0
Proteínas	7,1 g	9
Gorduras Totais	7,9 g	14
Gorduras Saturadas	5,0 g	23
Gorduras Trans	0,3 g	**
Fibra Alimentar	0	0
Sódio	263 mg	11
Cálcio	228 mg	23

* % Valores Diários de Referência com base em uma dieta de 2.000 kcal ou 8.400 kJ. Seus valores diários podem ser maiores ou menores dependendo de suas necessidades energéticas.

** VD não estabelecido.

Fig. 7. Acima, imagem com os traços de linha e coluna da imagem. Abaixo, imagem após tratamento de remoção de traços. Fonte: do autor.

Informações Nutricionais	Qtde.por porção*	%VD**
Valor energético	147kcal=605kJ	7%
Carboidratos	19 g	6%
Proteínas	2,0 g	3%
Gorduras totais	7,2 g	13%
Gorduras saturadas	3,7 g	17%
Gorduras trans	zero g	**
Gorduras Monoinsaturadas	2,1 g	**
Gorduras Polinsaturadas	0,5 g	**
Colesterol	zero mg	0%
Fibra alimentar	1,1 g	4%
Sódio	90 mg	4%
* %Valores Diários com base em uma dieta de 2.000 kcal ou 8400 kJ. Seus valores diários podem ser maiores ou menores dependendo de suas necessidades energéticas.		
** (%)VD não estabelecido.		

Informações Nutricionais	Qtde.por porção*	%VD**
Valor energético	147kcal=605kJ	7%
Carboidratos	19 g	6%
Proteínas	2,0 g	3%
Gorduras totais	7,2 g	13%
Gorduras saturadas	3,7 g	17%
Gorduras trans	zero g	**
Gorduras Monoinsaturadas	2,1 g	**
Gorduras Polinsaturadas	0,5 g	**
Colesterol	zero mg	0%
Fibra alimentar	1,1 g	4%
Sódio	90 mg	4%
* %Valores Diários com base em uma dieta de 2.000 kcal ou 8400 kJ. Seus valores diários podem ser maiores ou menores dependendo de suas necessidades energéticas.		
** (%)VD não estabelecido.		

Fig. 8. Acima, imagem com os traços de linha e coluna da imagem. Abaixo, imagem após tratamento de remoção de traços. Fonte: do autor.

Dessa forma, fez-se necessário o uso do modelo de aprendizado de máquina *K-means*, no qual foi utilizado apenas um $K = 2$, porque a cor de interesse era a cor de fundo a qual sempre prevalece em imagens de tabelas e o outro centróide fica responsável por coletar a cor da letra ou de outra característica.

Por fim, a função de *K-means* retorna as duas cores e sua porcentagem na imagem, de modo que a cor com predominância foi definida como a cor do traço que sobrescreveram as linhas e colunas da tabela (Fig. 9).

Informações Nutricionais	Qtde.por porção*	%VD**
Valor energético	147kcal=605kJ	7%
Carboidratos	19 g	6%
Proteínas	2,0 g	3%
Gorduras totais	7,2 g	13%
Gorduras saturadas	3,7 g	17%
Gorduras trans	zero g	**
Gorduras Monoinsaturadas	2,1 g	**
Gorduras Polinsaturadas	0,5 g	**
Colesterol	zero mg	0%
Fibra alimentar	1,1 g	4%
Sódio	90 mg	4%
* %Valores Diários com base em uma dieta de 2.000 kcal ou 8400 kJ. Seus valores diários podem ser maiores ou menores dependendo de suas necessidades energéticas.		
** (%)VD não estabelecido.		

Fig. 9. Imagem após tratamento de remoção de traços com uso de *K-means*. Fonte: do autor.

D. Redimensionamento e DPI

Por as imagens não terem um pré-processamento prévio das imagens e por não terem sido retiradas de um lugar unificado, foi necessário fazer um reescalonamento das mesmas, de modo que por padrão todas as imagens foram aumentadas em 4 vezes em seu tamanho padrão.

Todavia um aumento de 4 vezes também não foi suficiente, visto que algumas imagens já possuíam um tamanho padrão que seria superior às imagens aumentadas em 4 vezes. Dessa forma foi definido que todas as imagens teriam um valor de largura (do inglês, *width*) = a 1024 pixels, (o que equivale a $10\frac{43}{64}$ polegadas) em que depois, com o auxílio dos pacotes *Pillow* e *tempfile*, foi definido um valor de 300 DPIs para as imagens de tabela nutricional.

E. Máscara de nitidez

Para a máscara de nitidez, utilizou-se da equação 3 com o uso de um *kernel* de 3px. nas colunas da matriz e 3 px. nas linhas da matriz, assim, realizando a convolução com o uso de filtro Gaussiano. O valor de matriz 3 x 3 é o menor valor aceito para realizar a convolução com o filtro Gaussiano próprio da biblioteca OpenCV, dessa forma, não foi possível realizar testes com matriz menores que essa.

Além disso, o processo de máscara de nitidez contou com um parâmetro de sigma = 0,5, o que também é aplicado na função Gaussiana, e com o valor de *amount* = 1,5, o que pode ser representado por 100%. Por fim, o *threshold* foi definido como 0.

Informações Nutricionais	Qtde.por porção*	%VD**
Valor energético	147kcal=605kJ	7%
Carboidratos	19 g	6%
Proteínas	2,0 g	3%
Gorduras totais	7,2 g	13%
Gorduras saturadas	3,7 g	17%
Gorduras trans	zero g	**
Gorduras Monoinsaturadas	2,1 g	**
Gorduras Polinsaturadas	0,5 g	**
Colesterol	zero mg	0%
Fibra alimentar	1,1 g	4%
Sódio	90 mg	4%
* %Valores Diários com base em uma dieta de 2.000 kcal ou 8400 kJ. Seus valores diários podem ser maiores ou menores dependendo de suas necessidades energéticas.		
** (%)VD não estabelecido.		

Informações Nutricionais	Qtde.por porção*	%VD**
Valor energético	147kcal=605kJ	7%
Carboidratos	19 g	6%
Proteínas	2,0 g	3%
Gorduras totais	7,2 g	13%
Gorduras saturadas	3,7 g	17%
Gorduras trans	zero g	**
Gorduras Monoinsaturadas	2,1 g	**
Gorduras Polinsaturadas	0,5 g	**
Colesterol	zero mg	0%
Fibra alimentar	1,1 g	4%
Sódio	90 mg	4%
* %Valores Diários com base em uma dieta de 2.000 kcal ou 8400 kJ. Seus valores diários podem ser maiores ou menores dependendo de suas necessidades energéticas.		
** (%)VD não estabelecido.		

Fig. 10. Antes (acima) e depois (abaixo) da aplicação da máscara de nitidez. Fonte: do autor.

F. Dilatação

O processo de dilatação foi realizado com um *kernel* de 1 x 1, visto que acima disso os resultados ficavam tão embaçados que a dificuldade para o reconhecimento dos caracteres tornava-se difícil até mesmo para seres humanos.

Informações Nutricionais	Qtde.por porção*	%VD**
Valor energético	147kcal=605kJ	7%
Carboidratos	19 g	6%
Proteínas	2,0 g	3%
Gorduras totais	7,2 g	13%
Gorduras saturadas	3,7 g	17%
Gorduras trans	zero g	**
Gorduras Monoinsaturadas	2,1 g	**
Gorduras Polinsaturadas	0,5 g	**
Colesterol	zero mg	0%
Fibra alimentar	1,1 g	4%
Sódio	90 mg	4%
* %Valores Diários com base em uma dieta de 2.000 kcal ou 8400 kJ. Seus valores diários podem ser maiores ou menores dependendo de suas necessidades energéticas.		
** (%VD não estabelecido.		

Informações Nutricionais	Qtde.por porção*	%VD**
Valor energético	147kcal=605kJ	7%
Carboidratos	19 g	6%
Proteínas	2,0 g	3%
Gorduras totais	7,2 g	13%
Gorduras saturadas	3,7 g	17%
Gorduras trans	zero g	**
Gorduras Monoinsaturadas	2,1 g	**
Gorduras Polinsaturadas	0,5 g	**
Colesterol	zero mg	0%
Fibra alimentar	1,1 g	4%
Sódio	90 mg	4%
* %Valores Diários com base em uma dieta de 2.000 kcal ou 8400 kJ. Seus valores diários podem ser maiores ou menores dependendo de suas necessidades energéticas.		
** (%VD não estabelecido.		

Informações Nutricionais	Qtde.por porção*	%VD**
Valor energético	147kcal=605kJ	7%
Carboidratos	19 g	6%
Proteínas	2,0 g	3%
Gorduras totais	7,2 g	13%
Gorduras saturadas	3,7 g	17%
Gorduras trans	zero g	**
Gorduras Monoinsaturadas	2,1 g	**
Gorduras Polinsaturadas	0,5 g	**
Colesterol	zero mg	0%
Fibra alimentar	1,1 g	4%
Sódio	90 mg	4%
* %Valores Diários com base em uma dieta de 2.000 kcal ou 8400 kJ. Seus valores diários podem ser maiores ou menores dependendo de suas necessidades energéticas.		
** (%VD não estabelecido.		

Fig. 11. Tabela nutricional com *kernels* de 1x1, 2x2 e 3x3, respectivamente.
Fonte: do autor.

Todavia, atualmente encontra-se pesquisas que utilizam de um alto valor de dilatação, acompanhado a técnicas de encontrar contornos para fazer a detecção de textos (não reconhecimento) que destacam a ROI. Nessa área a rede de *deep learning* com maior desempenho, a qual é considerada estado da arte^[22], é a rede EAST.

G. Acurácia e tempo de execução

Por fim, após a realização de todo o *pipeline* para o tratamento das imagens, com um computador Intel(R) Core(TM) i5-4200U CPU @ 1.60GHz, 8049 MB de memória

RAM e com o sistema operacional Ubuntu 18.04.4 LTS, foram feitos os testes de desempenho do algoritmo, considerando que o mesmo não teve falhas, que comumente ocorre no processo de OCR, como citado na seção II.D.

Destaca-se que os resultados em que se aplicou o Tesseract OCR contou com a imagem sem tratamento algum (a qual tinha função de ser o parâmetro ao ser superado com os filtros), imagens na escala de cinza e imagens binarizadas, que tiveram uma diferença de 0,6 e 1,4 entre as médias de palavras e caracteres, respectivamente, em relação à detecção feita por um ser humano.

Tanto a contagem dos caracteres, quanto a contagem das palavras foram realizadas pelo *Libreoffice Writer*.

TABELA 1
Contagem De Palavras E Caracteres

	Humano		Tabela original sem tratamento		Tabela em escala de cinza		Tabela binarizada	
	Palavras	Caracteres	Palavras	Caracteres	Palavras	Caracteres	Palavras	Caracteres
Tabela 1	89	431	81	491	79	498	80	493
Tabela 2	89	497	75	403	79	419	80	419
Tabela 3	89	496	84	494	79	495	81	493
Tabela 4	111	801	61	397	90	510	90	515
Tabela 5	89	498	43	176	48	198	90	226
Tabela 6	84	425	79	445	82	460	80	458
Tabela 7	98	502	65	607	79	434	79	433
Tabela 8	50	269	78	434	78	446	78	445
Tabela 9	80	449	110	787	114	831	115	825
Tabela 10	84	445	87	505	84	492	84	492
Média	86,3	481,3	76,3	473,9	81,2	478,3	85,7	479,9
Desvio padrão	15,43480914	131,8711914	17,68269713	156,1775699	15,96384805	153,3630768	11,16592037	146,3963038

Além disso, vale pode-se observar uma diferença de 6 caracteres e 9,4 palavras entre as médias as das tabelas sem tratamento em relação à tabela com binarização. Por fim, com uma diferença de 1,6 caracteres e 4,5 palavras entre a média das imagens em escala de cinza e das imagens binarizadas.

Quanto ao tempo de execução da *pipeline* completa em todos os tratamentos, a que obteve melhor resultado foi com a tabela sem tratamentos, todavia, foi o que possuiu maior diferença quanto às médias de reconhecimento humano. Dessa forma, com uma diferença de cerca de 5,3s de diferença entre o método em que a imagem é binarizada e a sem tratamentos, na maioria das aplicações ainda compensa fazer a execução de toda a *pipeline*.

TABELA 2

Tempo De Execução De Todo O Processo (Em Segundos)

	Tabela original sem tratamento	Tabela em escala de cinza	Tabela binarizada
Tabela 1	3,164589882	3,44664073	9,407701254
Tabela 2	2,024025679	2,988069057	9,016830921
Tabela 3	2,467786551	2,993188381	8,829680443
Tabela 4	2,104414701	3,057608366	9,69769454
Tabela 5	1,598902225	3,019456625	9,999590397
Tabela 6	20,46795082	3,359797955	8,868143082
Tabela 7	7,141323805	3,40742445	8,621689796
Tabela 8	1,888143301	3,372760057	9,975107908
Tabela 9	1,735692263	4,001138926	11,28250265
Tabela 10	0,8340177536	2,97471261	10,65501523
Média	4,342684698	3,262079716	9,635395622
Desvio padrão	5,923644149	0,3255481034	0,863097076

IV. CONCLUSÃO

Conclui-se que com a aplicação dos filtros, foi possível ter uma melhora significativa, quanto ao processo de coleta de dados de tabela nutricional com o uso de imagens de tabelas que passa por todo o processo proposto, em que foi possível obter uma média de 85,7 palavras e 479,9 caracteres em um *dataset* de 10 amostras, enquanto os dados reconhecidos por um humano obteve uma média de 86,3 palavras e 481,3 caracteres no mesmo *dataset*.

REFERÊNCIAS

- [1] BCC Publishing. 2019 Food and Beverage Research Review. Disponível em: <https://www.bccresearch.com/market-research/food-and-beverage/food-and-beverage-research-review.html>. Acesso em: 18 de julho de 2020.
- [2] BRADSKI, Gary; KAEHLER, Adrian. Learning OpenCV: Computer vision with the OpenCV library. " O'Reilly Media, Inc.", 2008.
- [3] BRASIL. ANVISA/MS. RDC nº 40, de 21 de março de 2001. Aprova o Regulamento Técnico para Rotulagem Nutricional Obrigatória de alimentos e bebidas embaladas. Disponível em: <http://www.anvisa.gov.br>. Acesso em: 18 de julho de 2020.
- [4] BRASIL. ANVISA/MS. RDC nº 40, de 8 de fevereiro de 2002. Aprova o regulamento técnico para rotulagem de alimentos e bebidas embalados que contenham glúten. Disponível em: <http://www.anvisa.gov.br>. Acesso em: 18 de julho de 2020.
- [5] BROWN, Mark T. Automated Grading of Handwritten Numerical Answers. 2017.
- [6] DUNHAM, Margaret H. Data mining: Introductory and advanced topics. Pearson Education India, 2006.
- [7] EL HARRAJ, A.; RAISSOUNI, N. Toward indoor and outdoor surveillance using an improved fast background subtraction algorithm. International Journal of Computer, Control, Quantum and Information Engineering, v. 9, n. 4, p. 595-600, 2015.
- [8] GIODRY, Thad. Improving the quality of the output. Disponível em: <https://github.com/tesseract-ocr/tessdoc/blob/master/ImproveQuality.md>. Acesso em: 18 de julho de 2020
- [9] HARRAJ, Abdeslam El; RAISSOUNI, Naoufal. OCR accuracy improvement on document images through a novel pre-processing approach. arXiv preprint arXiv:1509.03456, 2015.
- [10] JULIASTUTI, E. et al. Image contrast enhancement for film-based dental panoramic radiography. In: 2012 International Conference on System Engineering and Technology (ICSET). IEEE, 2012. p. 1-5.
- [11] KAE, Andrew et al. Improving state-of-the-art OCR through high-precision document-specific modeling. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE, 2010. p. 1935-1942.
- [12] LEE, Che-Wei; TSAI, Wen-Hsiang. A secret-sharing-based method for authentication of grayscale document images via the use of the PNG image with a data repair capability. IEEE transactions on image processing, v. 21, n. 1, p. 207-218, 2011.
- [13] MAHMUD, Md Sohrab; RAHMAN, Md Mostafizer; AKHTAR, Md Nasim. Improvement of K-means clustering algorithm with better initial centroids based on weighted average. In: 2012 7th International Conference on Electrical and Computer Engineering. IEEE, 2012. p. 647-650.
- [14] OTSU, Nobuyuki. A threshold selection method from gray-level histograms. IEEE transactions on systems, man, and cybernetics, v. 9, n. 1, p. 62-66, 1979.
- [15] R. Fisher, S. Perkins, A. Walker and E. Wolfart. Unsharp filter. Disponível em: <https://homepages.inf.ed.ac.uk/rbf/HIPR2/unsharp.htm>. Acesso em: 20 de julho de 2020.
- [16] REUL, Christian et al. State of the Art Optical Character Recognition of 19th Century Fraktur Scripts using Open Source Engines. arXiv preprint arXiv:1810.03436, 2018.
- [17] RICE, Stephen V.; JENKINS, Frank R.; NARTKER, Thomas A. The fourth annual test of OCR accuracy. Technical Report 95, 1995.
- [18] SAUERBRONN, Ana Luiza Azambuja et al. Análise laboratorial da composição de alimentos processados como contribuição ao estudo da rotulagem nutricional obrigatória de alimentos e bebidas embalados no Brasil. 2003. Tese de Doutorado.
- [19] SHEPARD, David H. Character reading techniques. U.S. Patent n. 4,021,777, 3 maio 1977.
- [20] SMITH, Ray. An overview of the Tesseract OCR engine. In: Ninth international conference on document analysis and recognition (ICDAR 2007). IEEE, 2007. p. 629-633.
- [21] YANG, Hsueh-Yen; FAN, Yu-Cheng; TSAO, Hen-Wai. Algorithm and architecture design of adaptive brightness correction for video application. IEEE Transactions on Consumer Electronics, v. 54, n. 4, p. 1858-1866, 2008.
- [22] ZHOU, Xinyu et al. East: an efficient and accurate scene text detector. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. 2017. p. 5551-5560.
- [23] ZIMMERMAN, John B. et al. An evaluation of the effectiveness of adaptive histogram equalization for contrast enhancement. IEEE Transactions on Medical Imaging, v. 7, n. 4, p. 304-312, 1988.