

# Processamento de Imagens para Reconhecimento de Caracteres

Fellipe Carreiro de Oliveira e Lennon Ferreira Machado

*Ciência da Computação*

*Universidade Federal Rural do Rio de Janeiro*

*Nova Iguaçu, Rio de Janeiro, 26285–060*

*Email: flpo@ufrj.br; lennon@ufrj.br*

**Resumo**—O processamento de imagem tem um papel importante em inúmeras aplicações do dia a dia, da automação industrial ao reconhecimento facial. Neste estudo, exploraremos uma aplicação específica: o reconhecimento de texto em imagens, este que pode ser aplicado na digitalização de documentos, tradução automática e indexação de conteúdo visual, utilizando ferramentas e bibliotecas como Python, Tesseract e OpenCV. O relatório é dividido em cinco partes. A primeira parte apresenta o conceito de processamento de imagens. A segunda parte apresenta uma revisão das técnicas de pré-processamento. A terceira parte apresenta a proposta da solução do problema. A quarta parte apresenta os resultados obtidos. Finalmente, na quinta parte o que se conclui do experimento.

## 1. Introdução

Processamento de imagens é um conjunto de técnicas computacionais que visam a manipulação e análise de imagens digitais. O objetivo é extrair informações úteis das imagens, ou mesmo transformá-las para fins específicos. As técnicas de processamento de imagens que serão apresentadas são utilizadas para preparar as imagens para o reconhecimento de caracteres, bem como para melhorar a precisão do reconhecimento. As etapas básicas do processamento de imagens para reconhecimento de caracteres são as seguintes:

- Pré-processamento: o pré-processamento é utilizado para melhorar a qualidade da imagem e facilitar o reconhecimento de caracteres. As técnicas de pré-processamento incluem a correção de distorções, a remoção de ruídos e a segmentação da imagem.
- Reconhecimento de caracteres: o reconhecimento de caracteres é a etapa responsável por identificar os caracteres na imagem. As técnicas de reconhecimento de caracteres incluem o reconhecimento de caracteres isolados e o reconhecimento de caracteres conectados.

## 2. Técnicas Utilizadas

### 2.1. Tesseract

O OCR (Optical Character Recognition), reconhecimento óptico de caracteres, é uma técnica que possibilita a

extração de texto de documentos como imagens, fotografias, PDFs e diversos outros tipos de arquivos que contenham algum tipo de texto.

O Tesseract é um software de reconhecimento óptico de caracteres (OCR) de código aberto que pode ser usado para extrair texto de imagens. Ele é um dos softwares de OCR mais populares e versáteis disponíveis, e é usado em uma ampla variedade de aplicações, incluindo:

- Digitalização de documentos: o Tesseract pode ser usado para digitalizar documentos, como livros, revistas e jornais.
- Reconhecimento de placas de carro: o Tesseract pode ser usado para reconhecer placas de carro em imagens de câmeras de segurança.
- Reconhecimento de assinaturas: o Tesseract pode ser usado para reconhecer assinaturas em imagens de documentos.

No contexto de OCR, o Tesseract serve para converter imagens de texto em texto digital. Ele faz isso usando uma combinação de técnicas de processamento de imagens e um modelo de aprendizado de máquina para aprender a associar padrões de pixels a caracteres específicos.

### 2.2. Redução de ruído

O ruído é qualquer alteração na imagem que pode dificultar o reconhecimento de caracteres. As técnicas de redução de ruído são utilizadas para remover ou atenuar o ruído da imagem, melhorando sua qualidade.

Existem diversos tipos de filtros de redução de ruído, cada um com suas vantagens e desvantagens. Alguns tipos comuns incluem:

- Filtro de média: O filtro de média calcula o valor médio dos pixels vizinhos e atribui esse valor ao pixel central. Ele é eficaz na remoção de ruídos de fundo, como ruído de granulação e ruído de JPEG. No entanto, ele pode também suavizar a imagem, removendo detalhes importantes.
- Filtro de mediana: O filtro de mediana calcula a mediana dos pixels vizinhos e atribui esse valor ao pixel central. Ele é eficaz na remoção de ruídos de salt and pepper. No entanto, ele pode ser menos eficaz na remoção de outros tipos de ruído.

- Filtro de ponderação: O filtro de ponderação atribui pesos diferentes aos pixels vizinhos, com base na distância entre eles. Ele permite controlar a quantidade de ruído que é removida da imagem, ajustando os pesos dos pixels vizinhos.

### 2.3. Limiarização

Sua função permite binarizar as cores da imagem em branco e preto, sendo o limiar entre quais cores ficaram pretas ou brancas, parametrizável, assim como o seu tipo. No processo de limiarização, se a intensidade de um pixel na imagem de entrada for maior que o limiar especificado, o pixel correspondente na saída é marcado como branco. Se a intensidade do pixel de entrada for menor ou igual ao limiar, o pixel na saída é marcado como preto.

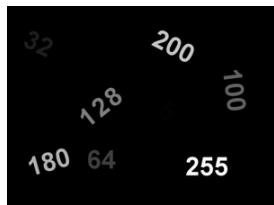


Figura 1. Imagem sem limiarização.

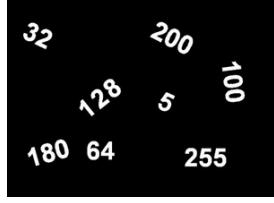


Figura 2. Imagem com limiarização.

Uma limitação da limiarização simples é a necessidade de especificar manualmente o valor do limiar. Isso pode ser tedioso e ineficaz na prática. Para superar esse problema, surge a necessidade de determinar automaticamente o limiar.

#### 2.3.1. Método de Otsu.

Algoritmos de limite global automático geralmente possuem as seguintes etapas.

- 1) Processar a imagem de entrada
- 2) Obtenha o histograma da imagem (distribuição de pixels)
- 3) Calcule o valor limite T
- 4) Substitua os pixels da imagem por branco nas regiões onde a saturação é maior que T e por preto nos casos opostos.

O método processa o histograma da imagem, segmentando os objetos pela minimização da variância em cada uma das classes. Geralmente, essa técnica produz resultados apropriados para imagens bimodais. O histograma de uma imagem desse tipo contém dois picos claramente expressos, representando diferentes intervalos de valores de intensidade. A ideia central consiste em separar o histograma da imagem em dois clusters, com um limiar definido como resultado da minimização da variância ponderada dessas classes, conforme indicado pelo método.

### 2.4. Erosão

A erosão é um procedimento que elimina pixels de uma imagem, reduzindo sua extensão e afinando os objetos. Isso pode ser empregado para eliminar regiões pequenas ou para desvincular objetos que estejam conectados.

A erosão é implementada por meio de uma operação de convolução com um kernel de erosão. O kernel de erosão é uma matriz quadrada, geralmente preenchida com números 1.

Para realizar a erosão, o kernel de erosão é aplicado à imagem pixel a pixel. Para cada pixel, o kernel é movido sobre a imagem, e o pixel atual é substituído pelo menor valor encontrado no kernel.

Por exemplo, considere a seguinte matriz:

$$\begin{matrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{matrix}$$

E o seguinte kernel de erosão:

$$\begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}$$

A dilatação da matriz com o kernel resultará na seguinte matriz:

$$\begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix}$$

Como pode ser visto, a dilatação aumentou o tamanho dos objetos na imagem.



Figura 3. Imagem normal sem técnica aplicada.



Figura 4. Imagem após a técnica de erosão.

### 2.5. Dilatação

A dilatação, o oposto da erosão, é usada para aumentar a área de objetos na imagem. Ela pode ser utilizada para preencher lacunas ou unir objetos próximos.

A dilatação é implementada por meio de uma operação de convolução com um kernel de dilatação. O kernel de dilatação é uma matriz quadrada, geralmente preenchida com números 1.

Para realizar a dilatação, o kernel de dilatação é aplicado à imagem pixel a pixel. Para cada pixel, o kernel é movido

sobre a imagem, e o pixel atual é substituído pelo maior valor encontrado no kernel.

Por exemplo, considere a seguinte matriz:

```
0 0 0  
0 1 0  
0 0 0
```

E o seguinte kernel de dilatação:

```
1 1 1  
1 1 1  
1 1 1
```

A dilatação da matriz com o kernel resultará na seguinte matriz:

```
1 1 1  
1 1 1  
1 1 1
```

Como pode ser visto, a dilatação aumentou o tamanho dos objetos na imagem.



Figura 5. Imagem normal sem técnica aplicada.



Figura 6. Imagem dilatada.

## 2.6. Abertura

A abertura é a combinação de erosão e dilatação. Ela é utilizada para remover pequenos detalhes da imagem, bem como pequenos buracos. O efeito básico de uma abertura é semelhante à erosão, pois tende a remover alguns dos pixels do primeiro plano (brilhantes) das bordas das regiões dos pixels do primeiro plano. No entanto, é menos destrutivo que a erosão em geral. Tal como acontece com outros operadores morfológicos, a operação exata é determinada por um elemento estruturante. O efeito do operador é preservar regiões de primeiro plano que tenham formato semelhante a este elemento estruturante, ou que possam conter completamente o elemento estruturante, enquanto elimina todas as outras regiões de pixels de primeiro plano.

## 2.7. Fechamento

O fechamento é a combinação de dilatação e erosão, respectivamente. Ele é utilizado para preencher pequenos

buracos na imagem, bem como pequenos detalhes. O fechamento assemelha-se, em alguns aspectos, à dilatação, visto que tende a expandir os limites das regiões em primeiro plano (brilhantes) em uma imagem, ao mesmo tempo que reduz os espaços vazios da cor de fundo nessas áreas. No entanto, destaca-se por preservar de maneira mais eficaz a forma do limite original. Similar a outros operadores morfológicos, a execução precisa do fechamento é determinada por meio de um elemento estruturante. O impacto desse operador consiste em manter intactas as regiões de fundo que compartilham características com o referido elemento estruturante ou que o contenham integralmente, ao passo que elimina todas as outras regiões de pixels de fundo.

Um dos usos da dilatação é preencher pequenos buracos de cor de fundo nas imagens, por exemplo, ‘ruído de pimenta’. Um dos problemas de fazer isso, entretanto, é que a dilatação também distorcerá todas as regiões dos pixels indiscriminadamente. Ao realizar uma erosão na imagem após a dilatação, ou seja, um fechamento, reduzimos parte desse efeito. O efeito do fechamento pode ser facilmente visualizado.

## 2.8. Correção de perspectiva

A correção de perspectiva é o processo de corrigir a distorção da imagem causada por uma perspectiva não ortogonal. Ela é utilizada para melhorar a precisão do reconhecimento de caracteres em imagens distorcidas. Comumente utilizada quando se quer digitalizar um documento.

No OpenCV, a correção de perspectiva é implementada por meio das funções `cv2.getPerspectiveTransform()` e `cv2.warpAffine()`.

A função `cv2.getPerspectiveTransform()` recebe como entrada quatro pontos nas imagens original e desejada. Esses pontos devem corresponder a objetos ou características que são visíveis em ambas as imagens. A função retorna uma matriz de transformação que pode ser usada para converter os pontos da imagem original para a imagem desejada.

A função `cv2.warpAffine()` recebe como entrada a matriz de transformação, a imagem original e a imagem desejada. A função aplica a transformação à imagem original e retorna a imagem transformada.

## 3. Sistema Proposto

O sistema OCR deste trabalho contou com os seguintes passos e suas respectivas funções:

### 3.1. Segmentação de texto

- 1) Filtro gaussiano para redução de ruído;
- 2) Limiarização adaptativa (OTSU);
- 3) Dilatação com kernel retangular;
- 4) Contorno, segmentação e ordenação;
- 5) Reconhecimento com Tesseract.

### **3.2. Segmentação de palavra**

- 1) Limiarização adaptativa (OTSU);
- 2) Dilatação com kernel retangular;
- 3) Contorno, segmentação e ordenação;
- 4) Reconhecimento com Tesseract.

### **3.3. Segmentação de caractere**

- 1) Filtro gaussiano para redução de ruído;
- 2) Filtro de nitidez;
- 3) Limiarização adaptativa (OTSU);
- 4) Abertura com kernel cruz;
- 5) Fechamento com kernel cruz;
- 6) Erosão com kernel cruz;
- 7) Contorno, segmentação e ordenação;
- 8) Reconhecimento com Tesseract.

## 4. Resultados

### 4.1. Imagem 1

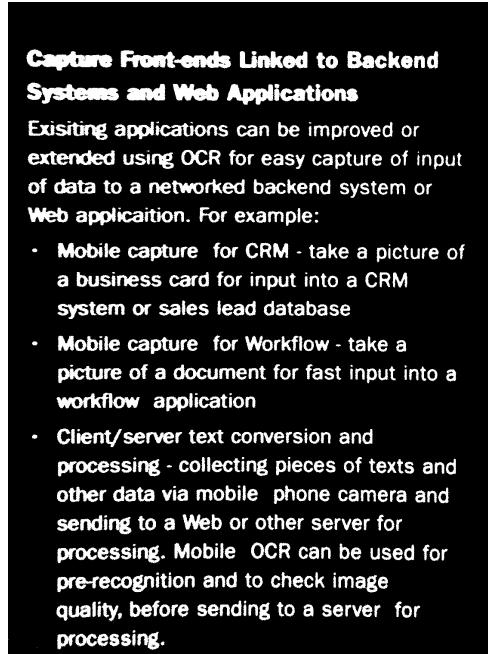


Figura 7. Imagem 1 - Resultado do pré-processamento.

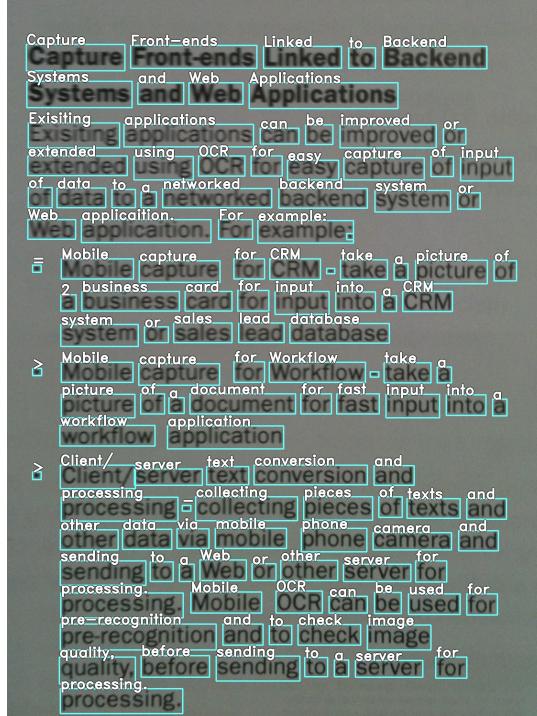


Figura 8. Imagem 1 - Resultado da segmentação de palavras e reconhecimento de caracteres.

Resultado 1. Imagem 1 - Extração do reconhecimento de caracteres.

### Capture Front-ends Linked to Backend Systems and Web Applications

Existing applications can be improved or extended using OCR for easy capture of input of data to a networked backend system or Web application. For example:

Mobile capture for CRM – take a picture of a business card for input into a CRM system or sales lead database

Mobile capture for Workflow – take a picture of a document for fast input into a workflow application

Client/server text conversion and processing – collecting pieces of texts and other data via mobile phone camera and sending to a Web or other server for processing. Mobile OCR can be used for pre-recognition and to check image quality, before sending to a server for processing.

## 4.2. Imagem 2

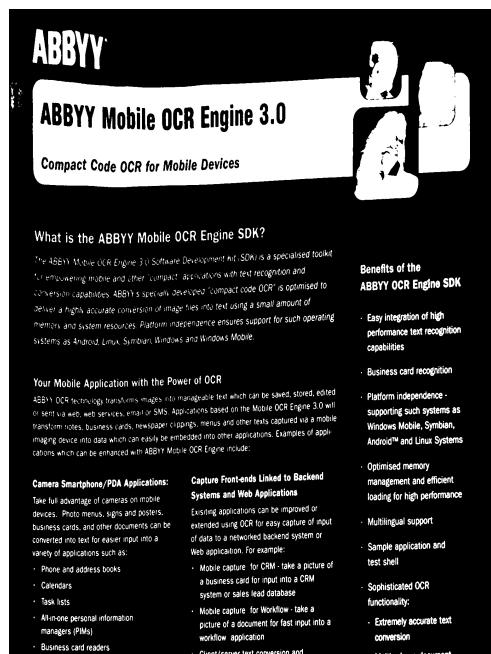


Figura 9. Imagem 2 - Resultado do pré-processamento.

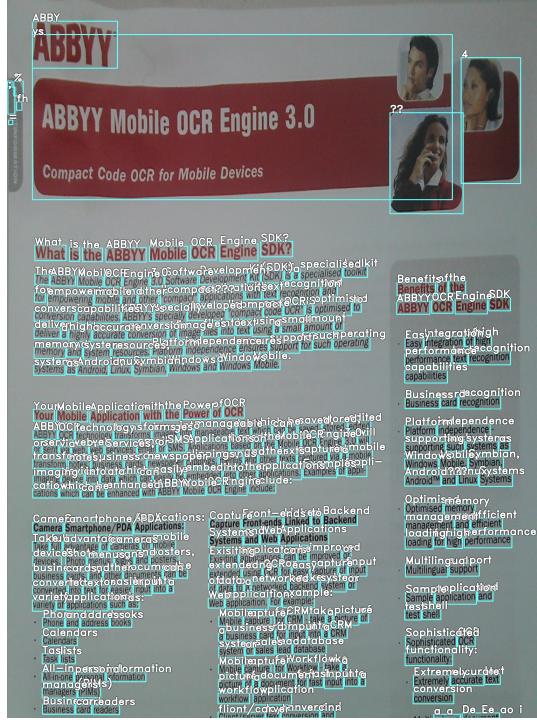


Figura 10. Imagem 2 - Resultado da segmentação de palavras e reconhecimento de caracteres.

Resultado 2. Imagem 2 - Extração do reconhecimento de caracteres.

ABBYY

ADDY Y

\>  
GUA Ter Coe ft  
mR EMO aU LM a Mac \}

What is the ABBYY Mobile OCR Engine SDK?

The ABBYY Mobile OCR Engine 3.0 Software Development Kit (SDK) is a specialised toolkit for empowering mobile and other "compact applications with text recognition and conversion capabilities". ABBYY's specially developed "compact code OCR" is optimised to deliver a highly accurate conversion of image files into text using a small amount of memory and system resources. Platform independence ensures support for such operating systems as Android, Linux, Symbian, Windows and Windows Mobile.

Benefits of the ABBYY OCR Engine SDK

- Easy integration of high performance text recognition capabilities

- Business card recognition

Your Mobile Application with the Power of OCR

ABBYY OCR technology transforms images into manageable text which can be saved, stored, edited or sent via web, web services, email or SMS. Applications based on the Mobile OCR Engine 3.0 will transform notes, business cards, newspaper clippings, menus and other texts captured via a mobile imaging device into data which can easily be embedded into other applications. Examples of applications which can be enhanced with ABBYY Mobile OCR Engine include:

Platform independence - supporting such systems as Windows Mobile, Symbian, Android and Linux Systems

+ Optimised memory management and efficient loading for high performance

Capture Front-ends Linked to Backend

Systems and Web Applications

Existing applications can be improved or extended using OCR for easy capture of input of data to a networked backend system or Web application. For example:

- + Mobile capture for CRM – take a picture of a business card for input into a CRM system or sales lead database
- + Mobile capture for Workflow – take a picture of a document for fast input into a workflow application

Atlant /earvar teyt panvereian and

Camera Smartphone/PDA Applications:

Take full advantage of cameras on mobile devices. Photo menus, signs and posters, business cards, and other documents can be converted into text for easier input into a variety of applications such as:

- + Phone and address books
- \* Calendars
- \* Task lists
- + All-in-one personal information

managers (PIMs)

\* Business card readers

Multilingual support

- Sample application and test shell
- + Sophisticated OCR functionality:
  - Extremely accurate text conversion

ava aie ek ee ee Oe ee

#### 4.3. Imagem 3

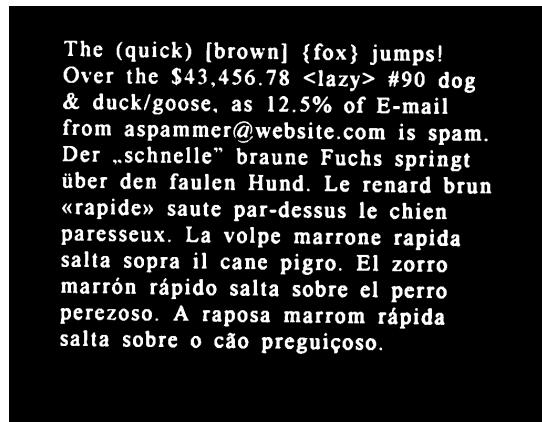


Figura 11. Imagem 3 - Resultado do pré-processamento.

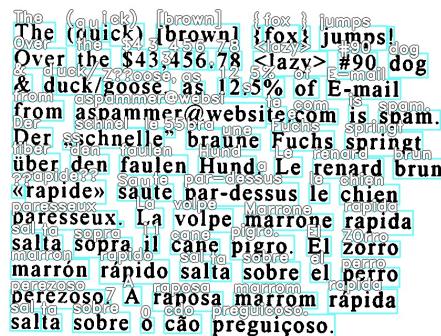


Figura 12. Imagem 3 - Resultado da segmentação de palavras e reconhecimento de caracteres.

Resultado 3. Imagem 3 - Extração do reconhecimento de caracteres.

The (quick) [ brown ] { fox } jumps !  
Over the \$43 ,456.78 <lazy> #90 dog  
& duck / goose , as 12.5% of E-mail  
from aspammer@website .com is spam .  
Der .. schnelle braune Fuchs springt  
iiber den faulen Hund. Le renard brun  
rapide saute par-dessus le chien  
paresseux. La volpe marrone rapida  
salta sopra il cane pigro. El zorro  
marron rapido salta sobre el perro  
perezoso. A raposa marrom rapida  
salta sobre o c o o preguicoso .

## 5. Conclusão

Os resultados deste trabalho demonstram a eficácia da técnica de pré-processamento de imagens proposta para a melhoria da precisão do reconhecimento de caracteres e palavras em imagens de câmera e scanner.

Em imagens de câmera (Imagen 1 e 2), a maioria dos caracteres foram localizados e reconhecidos corretamente. Porém, na imagem 2, foi observado que alguns caracteres não foram reconhecidos devido à estrutura da página e a presença de caracteres especiais.

Na imagem de scanner (Imagen 3), a maioria dos caracteres foi reconhecida corretamente.

Ao comparar o desempenho do OCR utilizando e não utilizando a técnica, nota-se que houve uma melhoria nos resultados em todas as imagens. A melhoria foi mais significativa em imagens com ruído, como a imagem 1 de câmera.

Além disso, foi observado que o Tesseract funciona muito melhor se for dado como entrada a palavra inteira pré-processada em vez do caractere individual. Também é importante dizer que após localizar os elementos, foram testadas diversas técnicas de tratamento da imagem, mas nos casos de exemplo, apenas a localização foi o suficiente para o reconhecimento.

As limitações dos resultados também foram mencionadas. No caso específico, a imagem 2 de câmera apresentou problemas de reconhecimento como anteriormente mencionado. Isso sugere que a técnica pode ser melhorada para lidar com esses tipos de imagens.

## Referências

- [1] OpenCV, *Image Processing in OpenCV*, [https://docs.opencv.org/4.x/d2/d96/tutorial\\_py\\_table\\_of\\_contents\\_imgproc.html](https://docs.opencv.org/4.x/d2/d96/tutorial_py_table_of_contents_imgproc.html)
- [2] OpenCV, *LearnOpenCV*, <https://learnopencv.com/otsu-thresholding-with-opencv/>
- [3] Nanonets, *OCR with Tesseract*, <https://nanonets.com/blog/ocr-with-tesseract/ocr-with-pytesseract-and-opencv>