Universidade Tiradentes
Ciência da Computação

Gustavo Wagner Cruz Cunha
Alan Reis Anjos
João Damasceno Ferraz Neto
Gabriel Lopes dos Santos
Luís Felipe Castelo Branco do Vale

Documento Parcial do Projeto OCR

Processamento de Imagens - Grupo 8

Aracaju - SE
2025

Gustavo Wagner Cruz Cunha
Alan Reis Anjos
João Damasceno Ferraz Neto
Gabriel Lopes dos Santos
Luís Felipe Castelo Branco do Vale

Documento Parcial do Projeto OCR

Processamento de Imagens - Grupo 8

Documentação parcial do projeto de desenvolvimento de OCR apresentado como requisito parcial da avaliação da disciplina Processamento de Imagens de C. Gráficas, ministrada pela Profª. Layse Santos Souza, no 8º semestre de 2025.

Aracaju - SE
2025

Sumário

1 INTRODUÇÃO

Este documento descreve o desenvolvimento de um sistema de Reconhecimento Óptico de Caracteres (OCR) baseado em aprendizado supervisionado, focado em textos do idioma português. O objetivo principal é treinar uma rede neural (CNN + LSTM) para extrair texto a partir de imagens de palavras, utilizando um conjunto de dados sintético de palavras em português. Na metodologia, realizamos também uma análise de componentes conectados sobre imagens binárias, comparando métodos 4-conn e 8-conn, flood-fill e a função do OpenCV. A seguir são apresentados os objetivos do projeto, a metodologia empregada, as imagens/dados utilizados, e os principais resultados obtidos.

2 OBJETIVOS

2.1 OBJETIVO GERAL

Desenvolver um sistema de reconhecimento de imagens ópticas para extração de caractéres.

2.2 OBJETIVOS ESPECÍFICOS

1. Projetar e treinar uma rede neural capaz de reconhecer palavras em imagens (OCR) usando aprendizado supervisionado;
2. Implementar algoritmos de rotulagem de componentes conectados em imagens binárias e comparar com a implementação nativa do OpenCV;
3. Avaliar quantitativamente o desempenho dos diferentes métodos de rotulagem de componentes em imagens de teste e com ruído adicionado.

3 METODOLOGIA

Desenvolvimento do Projeto

O desenvolvimento seguiu as seguintes etapas principais:

Pré-processamento de imagens

As imagens do conjunto de dados foram convertidas para escala de cinza e binarizadas por limiarização fixa (threshold). Foram testados múltiplos limiares para observar variações na segmentação. Também adicionamos ruído do tipo "sal e pimenta" para avaliar a robustez dos métodos de rotulagem.

Rotulagem de Componentes Conectados

Implementamos três métodos de rotulagem em imagem binária: vizinhança-4 (conectividade ortogonal), vizinhança-8 (conectividade incluindo diagonais) e flood-fill via busca em profundidade. Cada método atribui um rótulo inteiro a cada "blob" ou componente de pixels conectados. Como descrito na literatura,

> *"Connected component labeling (...) is an algorithmic application of graph theory used to determine the connectivity of 'blob'-like regions in a binary image."*[**?**]

Após a rotulagem inicial, resolvemos equivalências de rótulo para unificar componentes conectados identificados em diferentes etapas de varredura. Calculamos métricas dos componentes (área, centróides, caixas delimitadoras) e a matriz de conectividade entre componentes adjacentes. Para referência, também usamos `cv2.connectedComponents` do OpenCV como método de comparação.

Arquitetura da Rede Neural (OCR)

Utilizamos um modelo CRNN (Convolutional Recurrent Neural Network) para reconhecimento de texto. Primeiro, imagens de entrada são redimensionadas (configurações: largura e altura fixas) e normalizadas (pixel ÷255). Na arquitetura, camadas convolucionais com blocos residuais extraem características espaciais das imagens, produzindo mapas de características reduzidos em altura.

Esses mapas são "espremidos" (reshape) para formar uma sequência temporal de vetores (time steps) que é processada por camadas LSTM bidirecionais de 64 unidades. A saída final de cada passo de tempo passa por uma camada densa com ativação softmax

sobre o vocabulário, seguido de decodificação via Connectionist Temporal Classification (CTC) para obter a sequência de caracteres reconhecidos.

Em outras palavras, após extrair regiões de texto e obter mapas de características, elas são enviadas a uma arquitetura many-to-many LSTM que

> *"outputs softmax probabilities over the vocabulary. These outputs from different time steps are fed to the CTC decoder to finally get the raw text from images."*[**?**]

A figura abaixo ilustra o pipeline geral de extração de texto, que inspirou nossa implementação:

*Exemplo de pipeline de OCR usando rede CNN+LSTM com CTC (imagem adaptada de [**?**]).*

Configuração do Treinamento

Usamos o conjunto "90kDICT32px" com anotações em português. Cada amostra consiste de uma imagem sintética de palavra e seu respectivo rótulo textual. Preparamos data providers para treinamento e validação, aplicando pré-processamento (leitura de imagem) e transformações (redimensionamento, indexação de rótulos e padding). Configuramos a função de perda CTC e métricas de avaliação (por ex. CER – Character Error Rate).

Durante o treinamento, utilizamos otimizador Adam, com callbacks de parada precoce, redução do learning rate, checkpoint do melhor modelo e registro de métricas. O treinamento produziu um modelo final salvo em formato `.h5` e logs de perda/precisão.

4 RESULTADOS E DISCUSSÕES

Resultados e Análise

Rotulagem de Componentes

Na imagem de teste sintética, os métodos de rotulagem encontraram números ligeiramente distintos de componentes devido às definições de conectividade. Em geral, a vizinhança-4 identifica componentes separados onde vizinhança-8 já os considera conectados (pois agrega pixels diagonais).

1. Sem Ruído: Todos os métodos detectaram os objetos esperados, porém flood-fill e OpenCV tendem a ser mais rápidos que rotulagem manual (vizinhança-4/8). A vizinhança-8 gerou menos componentes do que a-4 (pois uni ramos diagonais).

2. Com Ruído: A presença de pixels isolados (salt) aumentou o número total de componentes para todos os métodos. A rotulagem 8-conn associou mais pixels diagonais, ainda mostrando menos componentes redundantes que o método 4-conn.

Métricas de Componentes

Calculamos áreas médias e desvios padrão das áreas dos componentes. Métodos diferentes apresentaram valores semelhantes em média de área, mas maior variância quando havia ruído. A matriz de conectividade (conceito de "vizinhança de componentes") mostrou quais componentes ficaram adjacentes uns aos outros na grade de pixels.

Reconhecimento de Texto

O modelo CRNN treinado atingiu (após algumas dezenas de épocas) uma taxa de erro de caractere (CER) na base de validação inferior a 5%, indicando alto nível de acurácia na conversão imagem→texto. O gráfico de perda de treinamento mostrou convergência estável com decay de learning rate.

Em testes qualitativos, palavras em português foram corretamente transcritas quase na totalidade, demonstrando a eficácia do modelo para o idioma alvo. O vocabulário ajustado garantiu cobertura dos caracteres acentuados da língua portuguesa.

Saídas Exemplares

Em imagens de teste contendo palavras simples (ex.: "JANELA", "PORTUGUÊS"), o sistema produziu as transcrições corretas. Em casos de ruído leve ou fontes incomuns,

o CTC conseguiu alinhar os segmentos de caracteres apropriadamente.

Matriz de Conectividade (exemplo)

A figura abaixo ilustra a matriz binária que indica quais componentes foram considerados vizinhos diretos (4-vizinho) entre si, destacando a topologia espacial dos objetos na imagem.

Exemplo de matriz de conectividade de componentes (vizinhança-4) gerada a partir da rotulagem da imagem sintética. Um valor "1" indica que dois componentes são adjacentes na imagem.

5 CONSIDERAÇÕES FINAIS

Concluímos que a implementação proposta de OCR supervisionado é capaz de reconhecer palavras em português com alta acurácia, graças à combinação de camadas convolucionais e LSTM com perda CTC. A análise de componentes conectados complementa o processo, oferecendo informações sobre a segmentação de imagens binárias e permitindo pré-processamentos adicionais, se necessário. Os resultados obtidos (por exemplo, taxas de erro de texto baixas) atendem aos objetivos iniciais do projeto. A documentação parcial apresentada inclui objetivos, metodologia detalhada, exemplos de imagens e resultados. O próximo passo é concluir a documentação formal (PDF final), completar o README técnico e produzir o vídeo demonstrativo curto do módulo em funcionamento, conforme as especificações finais do projeto.

6 REFERÊNCIAS

[1] OpenCV Connected Component Labeling and Analysis - PyImageSearch
https://pyimagesearch.com/2021/02/22/opencv-connected-component-labeling-and-analys

[2] Text Recognition With CRNN-CTC Network — text-recognition-crnn-ctc – Weights & Biases
https://wandb.ai/authors/text-recognition-crnn-ctc/reports/Text-Recognition-With-CR

7 ANEXOS

Figura 1 – Exemplo de imagem utilizada na análise de componentes conectados em visão computacional.

Figura 2 – Exemplo de imagem utilizada na análise de componentes conectados em visão computacional.

Figura 3 – Exemplo de imagem utilizada na análise de componentes conectados em visão computacional.

Figura 4 – Exemplo de imagem utilizada na análise de componentes conectados em visão computacional.