

ChatBot para Verificação de Documentos

Pedro Prado

Gabriel Mancini

Disciplina: Aprendizado Profundo - PPGCC

Resumo

Este relatório apresenta o desenvolvimento e funcionamento de um ChatBot destinado à verificação de documentos. O sistema utiliza técnicas de aprendizado profundo para processar entradas textuais e extrair informações relevantes, validando a conformidade e consistência dos dados. São detalhados os principais componentes do sistema, metodologias empregadas e resultados obtidos, bem como desafios enfrentados no processo de implementação.

Sumário

1	Introdução	2
2	Arquitetura do Sistema	2
3	Funcionamento do ChatBot	3
4	Implementação - ChatBot para RG	4
5	Implementação - ChatBot para Diploma	6
6	Resultados - Verificação de RG	9
7	Resultados - Verificação de Diploma	16
8	Desafios e Limitações	18
9	Conclusão	19

1 Introdução

Nos últimos anos, o aprendizado profundo emergiu como uma das áreas mais influentes da inteligência artificial, trazendo avanços significativos em visão computacional, processamento de linguagem natural e robótica. Este trabalho combina três componentes essenciais: aprendizado profundo, reconhecimento óptico de caracteres (OCR) com Tesseract e chatbots, para abordar desafios relacionados à automação de processos de verificação documental.

O aprendizado profundo, fundamentado em redes neurais artificiais, é particularmente eficaz na modelagem de padrões complexos em dados estruturados e não estruturados. De acordo com Goodfellow et al. (2016), modelos como redes neurais convolucionais (CNNs) e redes LSTM (Long Short-Term Memory) têm redefinido o estado da arte em reconhecimento de imagens e processamento de sequências textuais. Estas arquiteturas são amplamente utilizadas para tarefas que vão desde a classificação de imagens até a tradução automática de textos.

O Tesseract OCR, desde sua versão 4, integra redes LSTM como parte de seu pipeline de reconhecimento. Esse aprimoramento torna o Tesseract mais robusto em lidar com textos em fontes variadas e em contextos desafiadores, como documentos digitalizados de baixa qualidade (Smith, 2007). Além disso, o sistema permite a customização para melhorar o desempenho em cenários específicos, como a extração de informações de documentos oficiais.

Por outro lado, os chatbots têm se destacado como uma interface intuitiva entre usuários e sistemas computacionais. Segundo Brown et al. (2020), modelos baseados em aprendizado profundo, como transformers, possibilitam interações mais naturais e contextualizadas. Esses sistemas são particularmente úteis em aplicações como validação documental, onde o chatbot pode não apenas processar os dados, mas também interpretar e responder às consultas do usuário em tempo real.

Este trabalho explora a interação dessas tecnologias para desenvolver um sistema eficiente e automatizado de verificação de documentos. A abordagem proposta combina a robustez do OCR, a flexibilidade do aprendizado profundo e a interatividade dos chatbots, proporcionando uma solução integrada para os desafios de validação documental.

2 Arquitetura do Sistema

O ChatBot é composto por diferentes módulos que trabalham de forma integrada para realizar a verificação dos documentos. A seguir, são descritos os principais componentes:

Pré-processamento de Dados

O módulo de pré-processamento é responsável por:

- Carregar imagens de documentos (como RG) a partir de arquivos locais.
- Converter a imagem para escala de cinza, aplicar desfoque gaussiano para suavização e detectar bordas utilizando o algoritmo Canny.
- Localizar contornos na imagem para ajustar a perspectiva e melhorar a leitura dos caracteres.

Reconhecimento Óptico de Caracteres (OCR)

Para extrair as informações textuais dos documentos processados, o sistema utiliza o Tesseract OCR, que integra redes LSTM (Long Short-Term Memory) como parte de seu pipeline de reconhecimento. Este é um componente fundamental do domínio de aprendizado profundo, permitindo o reconhecimento eficiente de caracteres em textos digitalizados.

Interface do Usuário

A interface foi desenvolvida utilizando a biblioteca Gradio, integrando técnicas modernas de interação em linguagem natural. Esse módulo complementa o sistema, permitindo ao usuário interagir com o ChatBot de forma simples e intuitiva.

3 Funcionamento do ChatBot

O processo de verificação de documentos pelo ChatBot pode ser descrito nas seguintes etapas:

1. **Entrada de Dados:** O usuário envia uma imagem de documento ou realiza perguntas sobre informações extraídas do documento.
2. **Pré-processamento:** Técnicas de visão computacional, como suavização de imagens e detecção de bordas, são aplicadas para melhorar a qualidade da entrada.
3. **Extração e Reconhecimento de Texto (OCR):** O texto é extraído usando o Tesseract OCR. Esse processo utiliza redes LSTM para realizar o reconhecimento de caracteres, uma abordagem baseada em aprendizado profundo.
4. **Validação de Informações:** Os dados extraídos são analisados usando expressões regulares para identificar elementos como CPF e RG. Esses números são organizados e preparados para validação.
5. **Resposta ao Usuário:** A interface do chatbot retorna respostas personalizadas com base nas informações validadas e nas perguntas realizadas.

4 Implementação - ChatBot para RG

Nesta seção, detalhamos o código para a implementação de um ChatBot para verificação de RG, explicando cada etapa do processo.

Importação de Bibliotecas

O código inicia com a importação das bibliotecas necessárias:

```
1 import numpy as np
2 import cv2
3 import imutils
4 from google.colab.patches import cv2_imshow
5 from matplotlib import pyplot as plt
```

Essas bibliotecas permitem:

- Manipulação de arrays (NumPy);
- Processamento de imagens (OpenCV);
- Funções auxiliares para manipulação de imagens (imutils);
- Exibição de imagens no ambiente Google Colab (cv2_imshow);
- Visualização de gráficos (Matplotlib).

Montagem do Google Drive e Cópia de Arquivos

Os arquivos necessários são carregados do Google Drive:

```
1 from google.colab import drive
2 drive.mount('/content/gdrive')
3 !cp -R /content/gdrive/MyDrive/Cursos\ -\ recursos/OCR\ com\ Python\
    Imagens/Projeto2 imagens/
```

Esses comandos montam o Google Drive no ambiente Colab e copiam as imagens necessárias para o diretório de trabalho.

Função para Exibição de Imagens

Uma função personalizada foi criada para exibir imagens:

```
1 def mostrar(img):
2     fig = plt.gcf()
3     fig.set_size_inches(20, 10)
4     plt.axis("off")
5     plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
6     plt.show()
```

Esta função utiliza Matplotlib para exibir imagens convertidas para o formato RGB.

Pré-processamento da Imagem

A imagem do RG é carregada e processada:

```
1 img = cv2.imread('/content/imagens/RG.jpg')
2 original = img.copy()
3 mostrar(img)
4 (H, W) = img.shape[:2]
5 print(H, W)
```

A imagem é convertida para tons de cinza, suavizada e as bordas são detectadas:

```
1 gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
2 blur = cv2.GaussianBlur(gray, (5, 5), 0)
3 edged = cv2.Canny(blur, 60, 160)
4 mostrar(edged)
```

Essas etapas facilitam a detecção de contornos no próximo passo.

Detecção e Ordenação de Contornos

Os contornos são identificados e ordenados:

```
1 def encontrar_contornos(img):
2     cnts = cv2.findContours(img, cv2.RETR_LIST, cv2.CHAIN_APPROX_SIMPLE
3                             )
4     cnts = imutils.grab_contours(cnts)
5     cnts = sorted(cnts, key=cv2.contourArea, reverse=True)[:6]
6     return cnts
```

O contorno mais significativo, com quatro vértices, é selecionado para correção de perspectiva.

Correção de Perspectiva

A correção de perspectiva "achata" a imagem do RG:

```
1 matriz = cv2.getPerspectiveTransform(pts1, pts2)
2 transform = cv2.warpPerspective(original, matriz, (W, H))
3 mostrar(transform)
```

Isso ajusta a imagem para facilitar o reconhecimento de texto.

Reconhecimento Óptico de Caracteres (OCR)

O texto é extraído da imagem corrigida:

```

1 import pytesseract
2 texto = pytesseract.image_to_string(transform, lang="por", config=
    config_tesseract)
3 print(texto)

```

A biblioteca Tesseract é usada para reconhecimento óptico de caracteres (OCR).

Extração de Dados

Os números relevantes (CPF, RG) são extraídos usando expressões regulares:

```

1 import re
2 digitos = re.findall(r'\d', texto)
3 CPF = digitos[:11]
4 RG = digitos[11:19]

```

Esses números são processados e armazenados para validação.

Interface com ChatBot

Um chatbot é implementado para interação com o usuário:

```

1 def answer_question(question):
2     if question.lower() == "cpf":
3         return f"Ok, isto é um CPF valido: {texto}"
4     elif question.lower() == "rg":
5         return f"Ok, isto é um RG valido: {texto2}"

```

A interface é criada com a biblioteca Gradio, permitindo perguntas e respostas em tempo real:

```

1 with gr.Blocks() as app:
2     question_input = gr.Textbox(label="Digite sua pergunta")
3     answer_output = gr.Textbox(label="Resposta do chatbot", interactive=
    False)
4     question_input.submit(answer_question, inputs=question_input,
        outputs=answer_output)
5 app.launch()

```

5 Implementação - ChatBot para Diploma

Nesta seção, detalhamos a implementação de um ChatBot voltado para a validação de diplomas acadêmicos utilizando técnicas de OCR, visão computacional e processamento interativo. O sistema permite a extração de texto de imagens e PDFs de diplomas, bem como a detecção de termos específicos fornecidos pelo usuário.

Instalação e Configuração

Para garantir a funcionalidade do ChatBot, são realizadas as seguintes instalações e configurações:

```
1 !sudo apt install tesseract-ocr
2 !pip install pytesseract
3 !pip install wordcloud
4 !pip install spacy
5 !pip install pdf2image
6
7 # Baixar o modelo de linguagem para português
8 !mkdir tessdata
9 !wget -O ./tessdata/por.traineddata https://github.com/tesseract-ocr/
       tessdata/blob/master/por.traineddata?raw=true
10 !apt-get install tesseract-ocr-por -y
11 !python -m spacy download pt_core_news_sm
```

Esses comandos garantem a instalação do Tesseract OCR, a biblioteca Pytesseract para integração em Python, o modelo de linguagem em português e o suporte para processamento de PDFs com a biblioteca pdf2image.

Configuração do OCR

O diretório do modelo de linguagem é configurado para que o Tesseract possa reconhecer texto em português:

```
1 os.environ['TESSDATA_PREFIX'] = '/usr/share/tesseract-ocr/4.00/tessdata'
   /
2 config_tesseract = "/usr/share/tesseract-ocr/4.00/tessdata/"
```

Funções Auxiliares

Duas funções principais auxiliam na exibição e no processamento das imagens:

```
1 # Exibir a imagem no ambiente
2 def mostrar(img):
3     cv2_imshow(img)
4
5 # Processar texto usando OCR
6 def OCR_processa(img, config_tesseract):
7     texto = pytesseract.image_to_string(img, lang='por', config=
       config_tesseract)
8     return texto
```

A função `OCR_processa` utiliza o Tesseract para extrair texto da imagem, enquanto `mostrar` exibe o resultado processado.

Detecção e Marcação de Texto

O sistema permite a detecção de palavras-chave específicas no diploma e destaca visualmente as ocorrências:

```
1 def OCR_processa_imagem(img, termo_pesquisa, config_tesseract, min_conf,
2     fonte_dir):
3     resultado = pytesseract.image_to_data(img, config=config_tesseract,
4         lang='por', output_type=Output.DICT)
5     num_ocorrencias = 0
6     for i in range(0, len(resultado['text'])):
7         confianca = int(resultado['conf'][i])
8         if confianca > min_conf:
9             texto = resultado['text'][i]
10            if termo_pesquisa in texto:
11                x, y, img = caixa_texto(i, resultado, img, (0,0,255))
12                img = escreve_texto(texto, x, y, img, fonte_dir,
13                    (50,50,225), 14)
14                num_ocorrencias += 1
15
16    return img, num_ocorrencias
```

- `image_to_data`: Extrai informações detalhadas como texto, coordenadas de localização e níveis de confiança.
- `min_conf`: Filtra texto com baixa confiança no reconhecimento.
- `caixa_texto` e `escreve_texto`: Desenham caixas e adicionam texto sobre as áreas detectadas.

ChatBot para Validação

O ChatBot é projetado para interagir com o usuário e processar diplomas em imagens ou PDFs. Ele permite:

- Upload do diploma para validação.
- Pesquisa por palavras-chave fornecidas pelo usuário.
- Retorno de resultados em tempo real, com destaque visual das ocorrências encontradas.

```
1 def chatbot_validador():
2     print("Bem-vindo ao validador de diplomas!")
3
4     # Upload do arquivo
5     uploaded = files.upload()
6
```

```

7     # Termo de pesquisa fornecido pelo usuario
8     termo_pesquisa = input("Digite o termo que deseja validar no diploma
9         : ")
10
10    for arquivo_nome in uploaded.keys():
11        if arquivo_nome.lower().endswith('.pdf'):
12            paginas = convert_from_path(arquivo_nome)
13            total_ocorrencias = 0
14            for i, pagina in enumerate(paginas):
15                img = np.array(pagina)
16                img, num_ocorrencias = OCR_processa_imagem(img,
17                    termo_pesquisa, config_tesseract, min_conf=2,
18                    fonte_dir='/content/calibri.ttf')
19                total_ocorrencias += num_ocorrencias
20                mostrar(img)
21            print(f'{termo_pesquisa} encontrado {total_ocorrencias}
22                  vez(es) no PDF.')
23
24    else:
25        img = cv2.imread(arquivo_nome)
26        img, num_ocorrencias = OCR_processa_imagem(img,
27            termo_pesquisa, config_tesseract, min_conf=2, fonte_dir
28            ='./content/calibri.ttf')
29        mostrar(img)
30        print(f'{termo_pesquisa} encontrado {num_ocorrencias} vez(
31            es) na imagem.')

```

O ChatBot utiliza a biblioteca `files` para upload de arquivos e integra o processamento de OCR e validação de texto, permitindo a interação eficiente com o usuário.

6 Resultados - Verificação de RG

Nesta seção, apresentamos os resultados detalhados obtidos ao longo do processo de verificação de um documento de identidade (RG) utilizando o sistema implementado. Cada etapa do processo é ilustrada e explicada em detalhes, demonstrando as funcionalidades e potenciais do sistema, desde a entrada do documento até a validação final por meio do chatbot.

Imagen Original do Documento



Figura 1: Imagem com várias tarjas pretas.

A Figura 1 apresenta a imagem original do RG, utilizada como entrada no sistema. Esse documento contém informações relevantes, como número do RG, CPF, data de emissão e assinatura. A qualidade dessa imagem é essencial, pois influencia diretamente na eficiência do processamento e na precisão da extração de dados.

Conversão para Escala de Cinza

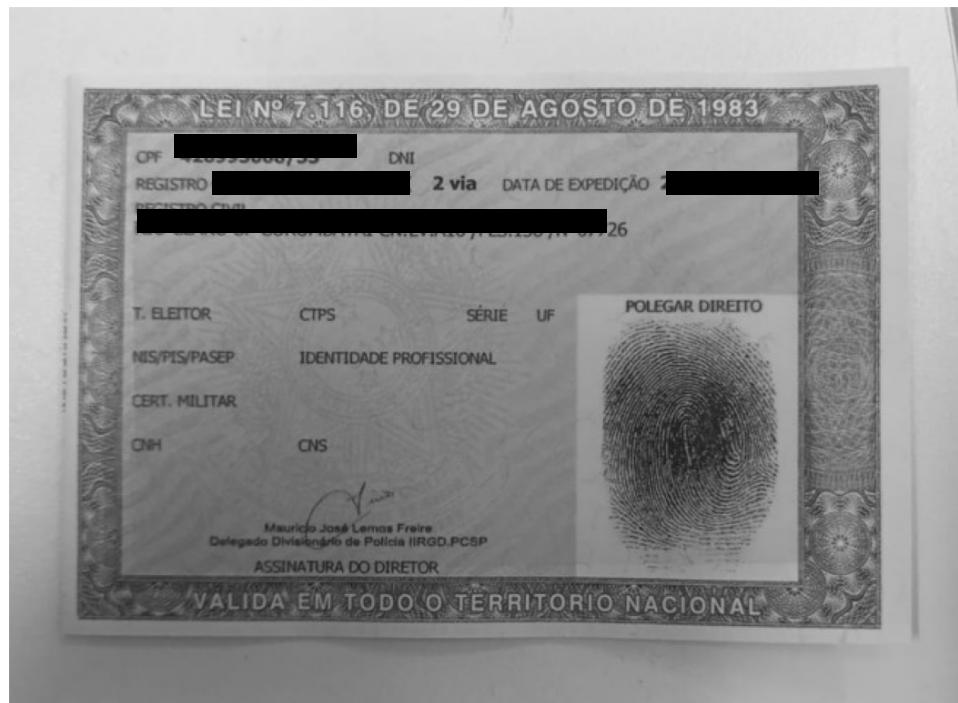


Figura 2: Imagem convertida para tons de cinza.

A Figura 2 mostra a imagem original após a conversão para escala de cinza. Essa etapa elimina informações de cor que não são necessárias para o reconhecimento óptico de caracteres (OCR), simplificando a complexidade visual. A maior distinção entre os elementos textuais e o fundo melhora a detecção de padrões textuais.

Aplicação de Desfoque Gaussiano



Figura 3: Imagem suavizada com desfoque gaussiano.

A Figura 3 apresenta o resultado da aplicação do desfoque gaussiano. Essa técnica reduz ruídos e detalhes irrelevantes, suavizando a imagem e destacando as bordas dos caracteres. Isso é crucial para minimizar interferências na próxima etapa, que envolve a detecção de bordas.

Detecção de Bordas com Canny

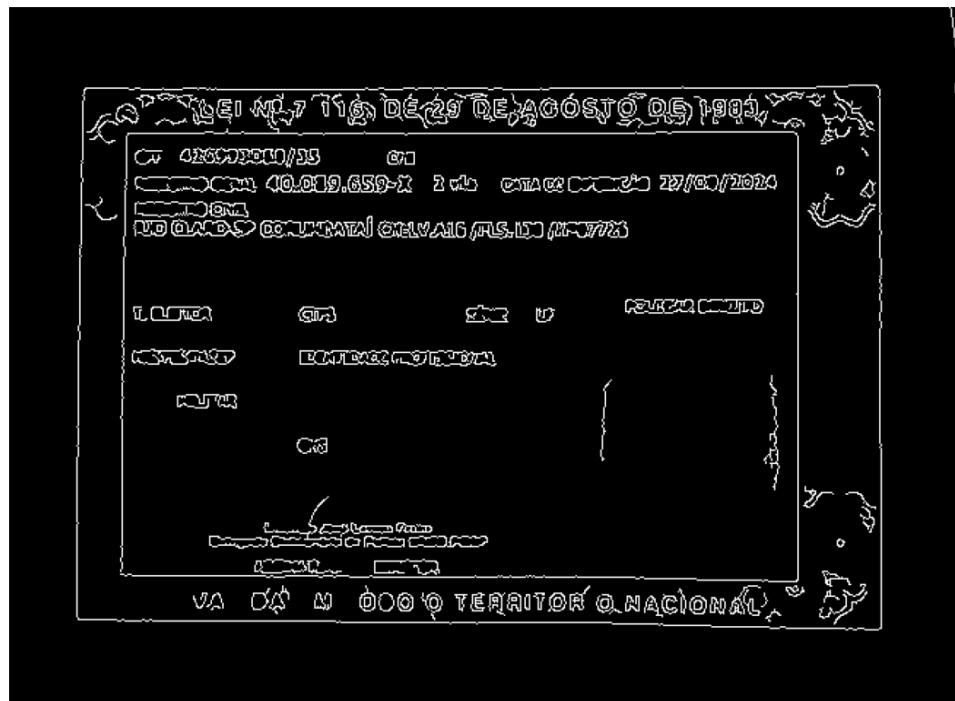


Figura 4: Detecção de bordas utilizando o algoritmo Canny.

A Figura 4 exibe o resultado da detecção de bordas, realizada utilizando o algoritmo de Canny. Essa etapa identifica os contornos mais significativos da imagem, permitindo a delimitação precisa das áreas de interesse, como campos de texto ou outras seções relevantes do documento.

Identificação e Correção de Perspectiva

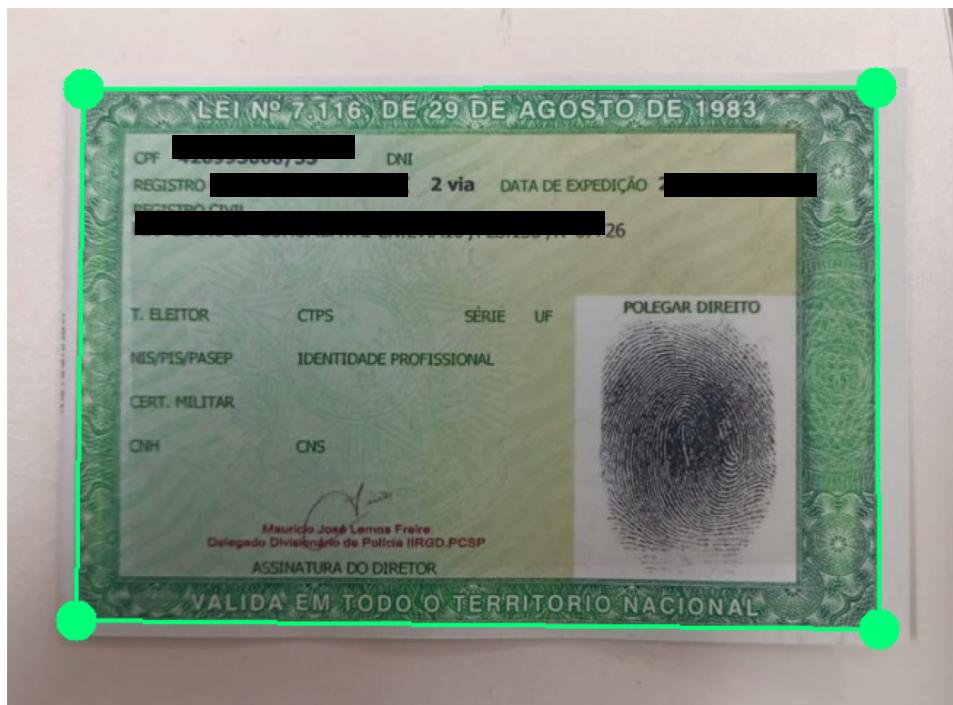


Figura 5: Imagem corrigida para ajuste de perspectiva.

A Figura 5 ilustra a imagem após a correção de perspectiva. Essa transformação é crucial para alinhar o documento, especialmente quando ele foi fotografado em ângulos inclinados. O alinhamento uniforme dos campos de texto facilita a extração de dados pelo OCR, garantindo maior precisão.

Extração e Validação com Gradio

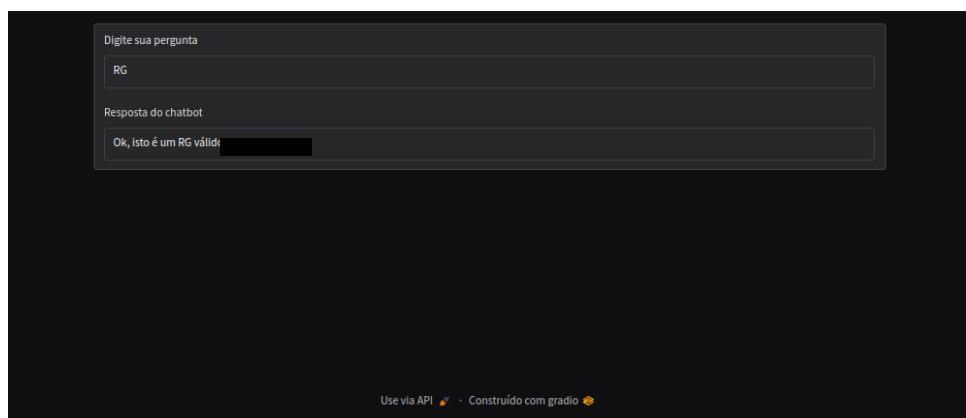


Figura 6: Interface Gradio para validação de RG.

A última etapa envolve a extração do texto com o Tesseract OCR e sua validação por meio da interface Gradio. Na Figura 6, observa-se a interface interativa onde o

usuário insere consultas (como "RG"), e o sistema valida os dados extraídos, retornando informações relevantes sobre o documento.

Possíveis Aplicações na UNESP

A integração desse sistema no ambiente administrativo da UNESP pode trazer inúmeras vantagens, agilizando processos e reduzindo a necessidade de intervenções manuais. Destacam-se as seguintes aplicações:

1. Validação de Documentos Acadêmicos

O sistema pode ser utilizado para verificar automaticamente documentos de identidade em processos administrativos, como matrícula de novos alunos, inscrição em programas de bolsas e submissão de documentos em plataformas digitais. Isso economiza tempo e reduz erros humanos.

2. Atendimento Digital

Integrado aos sistemas de atendimento ao aluno, o chatbot pode realizar a validação de documentos enviados online, permitindo respostas rápidas e automatizadas para consultas sobre validade e autenticidade de informações.

3. Processos de Pesquisa

Em atividades de pesquisa que exijam identificação e validação de participantes, o sistema pode agilizar a verificação de documentos, garantindo maior eficiência e confiabilidade nos dados coletados.

4. Organização de Eventos Acadêmicos

Em congressos e simpósios organizados pela UNESP, o sistema pode ser empregado para validar documentos de identidade de participantes, automatizando a conferência e acelerando o credenciamento.

Benefícios Gerais

A implementação do sistema na UNESP proporcionaria:

- Maior eficiência em processos administrativos.
- Redução de erros na validação de documentos.
- Atendimento ágil e acessível para a comunidade acadêmica.

- Redução da carga de trabalho dos funcionários, permitindo maior foco em atividades estratégicas.

Os resultados apresentados demonstram a robustez do sistema em processar e validar documentos de identidade, evidenciando sua aplicabilidade em instituições como a UNESP para modernizar processos administrativos e acadêmicos, trazendo eficiência e segurança. Contudo, para avanços significativos na aplicação prática, é fundamental a construção de um banco de dados contendo os documentos dos estudantes. Esse banco de dados permitirá refinar o treinamento do sistema, garantindo maior precisão e adaptabilidade a variações reais encontradas nos documentos.

Análise Final

Os resultados demonstram a eficácia do sistema em processar imagens de documentos de identidade. A combinação de técnicas de pré-processamento, detecção de bordas e OCR permitiu a extração precisa do texto. A integração com a interface Gradio facilitou a interação com o usuário, proporcionando uma experiência intuitiva e eficiente.

Os principais desafios identificados foram:

- Qualidade da imagem original: Imagens de baixa qualidade podem afetar o desempenho do OCR.
- Complexidade visual: Elementos como fundos texturizados e marcas d'água podem introduzir ruído.

Apesar dessas limitações, o sistema demonstrou robustez em condições variadas. Entretanto, para melhorar ainda mais seu desempenho e garantir uma aplicação mais abrangente, é imprescindível a criação de um banco de dados com os documentos dos estudantes. Isso possibilitará a personalização do modelo, aumentando a eficiência e a confiabilidade do sistema em cenários reais.

7 Resultados - Verificação de Diploma

Nesta seção, apresentamos os resultados obtidos ao aplicar o sistema de validação no documento de diploma fornecido. O objetivo foi verificar a capacidade do sistema em identificar palavras específicas no documento utilizando técnicas de OCR (Reconhecimento Óptico de Caracteres) e visão computacional.

Documento Original

A imagem original do diploma foi carregada no sistema como entrada para o processo de validação. O diploma contém informações textuais importantes, como o nome da

instituição, o nome do aluno, a data de emissão e o número de registro, além de elementos decorativos que adicionam complexidade ao documento.

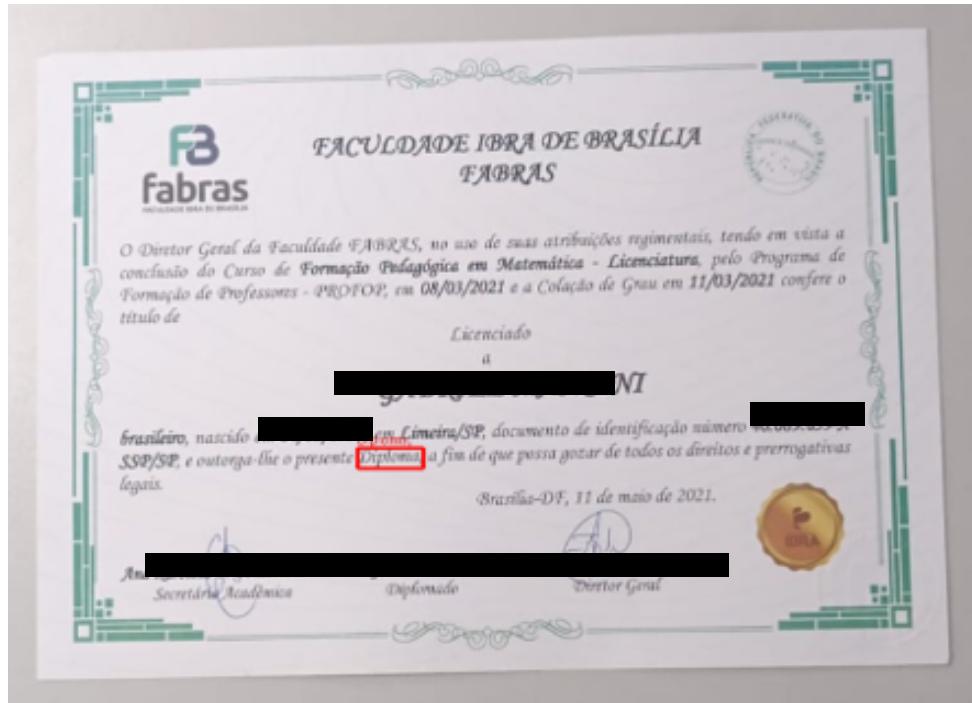


Figura 7: Imagem original do diploma.

A Figura 7 apresenta o diploma fornecido, que foi utilizado como base para o processo de validação.

Resultados da Validação

Durante o processamento da imagem, o sistema realizou pré-processamento (conversão para escala de cinza, suavização, detecção de bordas e correção de perspectiva) e aplicou OCR para extrair o texto. O principal resultado foi:

- A palavra "**diploma**" foi corretamente identificada no corpo principal do texto, indicando que o sistema conseguiu processar parcialmente o conteúdo.

No entanto, informações adicionais, como nomes ou datas, não foram reconhecidas com precisão devido às limitações descritas a seguir.

Limitações Observadas

O teste revelou algumas limitações ao processar o diploma:

- **Complexidade Visual:** Elementos decorativos, selos e bordas introduzem ruídos que dificultam a identificação precisa das palavras.

- **Qualidade da Imagem:** Distorções causadas por iluminação desigual ou reflexos afetaram a precisão do OCR, especialmente em textos menores.
- **Estilo da Fonte:** Fontes estilizadas ou decorativas foram mais difíceis de reconhecer, resultando em falhas.
- **Parâmetros do OCR:** As configurações padrão do Tesseract podem não estar otimizadas para este tipo de documento.

Análise Final

O sistema mostrou-se funcional ao identificar palavras específicas, como "**diploma**", mas apresentou limitações na identificação de informações detalhadas, evidenciando a necessidade de melhorias no pré-processamento e na configuração do OCR.

Possíveis Melhorias

Com base nos resultados, sugerimos:

- **Treinamento de Modelos Específicos:** Utilizar redes neurais treinadas para diplomas nacionais para melhorar a identificação de padrões textuais.
- **Aprimoramento do Pré-processamento:** Implementar técnicas adicionais de segmentação e remoção de ruído para isolar melhor as áreas textuais.
- **Ajustes no OCR:** Otimizar os parâmetros do Tesseract para adequar-se às características dos diplomas.
- **Uso de Redes Convolucionais:** Incorporar CNNs para segmentação e reconhecimento de texto, complementando o OCR tradicional.

8 Desafios e Limitações

Durante o desenvolvimento e implementação do sistema de verificação de RG, CPF e Diploma, foram identificados diversos desafios que impactaram seu desempenho. Entre os principais, destacam-se:

- **Qualidade da Imagem Original:** Documentos digitalizados ou fotografados em condições adversas, como baixa iluminação, sombras ou baixa resolução, dificultaram o reconhecimento de caracteres pelo OCR. Isso foi particularmente evidente em textos pequenos ou desgastados, como em RGs e CPFs mais antigos.

- **Complexidade Visual dos Documentos:** Elementos decorativos, marcas d'água e fundos texturizados presentes nos diplomas e em algumas versões de RGs introduziram ruídos que afetaram a detecção de bordas e a extração precisa do texto. Esses fatores aumentaram a dificuldade do OCR em interpretar corretamente as informações.
- **Variações de Perspectiva:** Fotografias tiradas em ângulos inadequados ou com distorções exigiram uma etapa de correção de perspectiva. Embora eficiente, essa correção depende da qualidade da detecção de contornos, que pode ser prejudicada em documentos com bordas desgastadas ou mal definidas.
- **Configuração do OCR:** O Tesseract requer ajustes específicos, como idioma e nível de confiança mínima, para otimizar o reconhecimento. Configurações inadequadas comprometeram a precisão em documentos com formatações atípicas ou fontes não padrão.
- **Limitações de Dados:** O conjunto de exemplos utilizados para teste foi limitado, o que impactou o desempenho em documentos com formatos ou fontes pouco comuns, como diplomas personalizados.
- **Dependência de Interação Humana:** Apesar de a interface Gradio facilitar a interação, ainda há dependência do usuário fornecer informações de pesquisa e interpretar os resultados, limitando a automação completa.

Embora esses desafios não inviabilizem o sistema, evidenciam a necessidade de aprimoramentos, especialmente na robustez do OCR e no pré-processamento das imagens.

9 Conclusão

O sistema de verificação de RG, CPF e Diploma, desenvolvido com aprendizado profundo, técnicas de visão computacional e uma interface interativa, mostrou-se uma solução promissora para automação de processos de validação documental. O sistema integra:

- Técnicas de pré-processamento de imagens, como conversão para escala de cinza, suavização, detecção de bordas e correção de perspectiva.
- Reconhecimento Óptico de Caracteres (OCR) com o Tesseract, auxiliado por redes LSTM, para extração confiável de texto.
- Interface Gradio, proporcionando uma interação intuitiva e validação em tempo real.

Os resultados demonstraram precisão satisfatória em imagens de boa qualidade e documentos padronizados, destacando a eficácia do sistema em identificar elementos essenciais,

como números de RG e CPF, além da palavra-chave "diploma" em certificados. A aplicabilidade prática inclui uso em matrículas, validações administrativas e eventos acadêmicos na UNESP.

Entretanto, os desafios enfrentados, como ruídos visuais e variações na qualidade dos documentos, apontam para melhorias futuras:

- Treinamento de modelos OCR específicos para padrões nacionais de RG, CPF e diplomas.
- Implementação de redes neurais convolucionais (CNNs) para segmentação e reconhecimento de texto.
- Automação completa do fluxo de validação, minimizando interações manuais.

Com base nos resultados, o sistema apresenta uma base sólida para expansões futuras, consolidando-se como uma ferramenta eficiente para processos administrativos e acadêmicos que envolvam validação de documentos em instituições como a UNESP e além.

Referências

- GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. *Deep Learning*. MIT Press, 2016.
- SMITH, Ray. An Overview of the Tesseract OCR Engine. *Proceedings of the Ninth International Conference on Document Analysis and Recognition*, IEEE, 2007.
- BROWN, Tom et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 2020.