Uso de Reconhecimento Óptico de Caracteres para Extração de Textos em Imagens de Redações

Filipe A. Sampaio[1], **Raimundo S. Moura**[1], **Kelson R. T. Aires**[1]

[1] Departamento de Computação – Universidade Federal do Piauí (UFPI)

felipealvessampaio@hotmail.com, rsm@ufpi.edu.br, kelson@ufpi.edu.br

Abstract. *Automatic Essay Scoring is a task in the area of Natural Language Processing, whose objective is to evaluate and score written prose texts. One of the main difficulties of this task is the lack of datasets of essays annotated with the value obtained in each competence. Thus, this work proposes an effective solution to capture essays written by students, through computer vision and optical character recognition techniques. This paper segments words from the image of the essay text and processes each word, then recognizing the text of each image. At the end, it orders all the words in the correct reading sequence, obtaining moderate performance.*

Resumo. *Avaliação Automática de Redação é uma tarefa da área de Processamento de Linguagem Natural, cujo objetivo é avaliar e pontuar textos em prosa escrita. Uma das principais dificuldades desta tarefa é a deficiência de datasets de redações anotadas com o valor obtido em cada competência. Assim, este trabalho propõe uma solução eficaz para capturar as redações escritas por alunos, através de técnicas de visão computacional e reconhecimento óptico de caracteres. Esse trabalho segmenta palavras da imagem do texto da redação e processa cada palavra, reconhecendo então o texto de cada imagem. Ao final, ordena todas as palavras na sequência correta da leitura, obtendo desempenho moderado.*

1. Introdução

Atualmente, no Brasil, o Exame Nacional do Ensino Médio (ENEM) possui a maior prova de redação do país em termos de participantes [INEP 2021]. Na edição de 2021 houve mais de 3,3 milhões de inscrições, com 2,1 milhões realizando a prova. Apenas 22 candidatos obtiveram nota 1.000 na redação do exame, com 4,57% dos participantes zerando a nota de redação. No ano anterior, 28 candidatos alcançaram nota máxima, com 3,22% obtendo nota mínima.

Observa-se então que nos últimos anos o desempenho das notas de redação vem caindo e o número de notas zeradas vem aumentando. Segundo [Barros 2019], parte do problema que traz tais resultados está relacionado ao baixo volume de produções textuais dos candidatos durante o período de estudos para o exame, além da falta de leitura por parte dos mesmos. A evolução do desempenho dos candidatos para elaboração de boas redações está ligada diretamente também ao *feedback* que professores e profissionais da área de Letras/Língua Portuguesa fornecem ao corrigir seus textos, informando o que deve ser melhorado.

Nota-se também um aumento na sobrecarga do trabalho desses profissionais, que acompanham os alunos durante todo o período de preparação para o exame. Para auxiliar nesse problema, há serviços onde os candidatos digitam suas redações e submetem em formulários para avaliação por um profissional. Um exemplo é o UOL Brasil Escola[1], que todo mês disponibiliza um tema de redação diferente para os alunos. A redação é corrigida manualmente e o aluno pode acompanhar sua evolução com suas notas recebidas.

Há também protótipos de ferramentas para correção automática de redações. Essas ferramentas são projetadas para auxiliar o profissional na correção, diminuindo sua carga de trabalho e otimizando o acompanhamento mais eficiente para com o aluno. Um exemplo desse tipo de ferramenta é o AAREM (Avaliador Automático de Redações para o Ensino Médio) [Marinho et al. 2022], que apresenta estratégias para avaliação automática de redações escritas em português por meio de uma abordagem baseada na definição de *features* e modelos específicos para cada competência da matriz de referência do ENEM.

Observa-se também que geralmente na proposta de correção manual, o candidato submete uma imagem da redação que foi escrita a mão. Assim, o tempo da correção e retorno do *feedback* está ligado diretamente na qualidade da imagem submetida e da escrita do aluno. Já a proposta de correção por ferramentas automáticas depende que o candidato digite seu texto em um formulário, prejudicando o mesmo na prática da escrita em folha. A submissão de texto para as ferramentas automáticas de correção é um processo muito valioso para os sistemas de NLP (*Natural Language Processing* - Processamento de Linguagem Natural), pois segundo [Marinho et al. 2021] há poucos *datasets* disponíveis para treinar modelos de correção.

Logo, é de grande importância que o candidato tenha a prática da escrita como uma constante em seus estudos diários. Assim também é necessário um *feedback* das correções textuais a partir do uso de ferramentas de correção, que pode otimizar o trabalho do profissional. A proposta deste trabalho é possibilitar essa ligação, oferecendo uma metodologia para extrair textos de imagens e posterior submissão aos sistemas de correção. Tal metodologia utiliza de técnicas de visão computacional e reconhecimento óptico de caracteres através de redes neurais. Atualmente, o principal problema dessa metodologia está ligada à qualidade da imagem e da escrita do candidato.

O restante deste trabalho está organizado nas seguintes seções: na Seção 2 são citados os principais trabalhos selecionados sobre o assunto abordado; na Seção 3 é apresentada a proposta metodológica utilizada; a Seção 4 apresenta os experimentos realizados; na Seção 5 são discutidos os resultados obtidos nos experimentos; por fim, a Seção 6 apresenta a conclusão e os trabalhos futuros.

2. Trabalhos Relacionados

Existe na literatura alguns trabalhos que procuram estudar OCR (*Optical Character Recognition* - Reconhecimento de Caractere Optico) e ICR (*Intelligent Character Recognition* - Reconhecimento Inteligente de Caracteres) ligado a Língua Portuguesa, onde definem métodos novos ou sugerem melhorias em modelos atuais. Observa-se também que há poucos trabalhos que focam no estudo de metodologias de extração de caracteres manuscritos no formato *off-line*, principalmente ligado também à língua portuguesa.

[1]https://brasilescola.uol.com.br/

Em [Zhou et al. 2017] os autores propuseram um método de detecção de texto de cena que consiste em dois estágios: uma rede totalmente convolucional e um estágio de fusão NMS (*Non-Maximum Suppression* - Supressão não máxima). A rede totalmente conectada produz regiões de texto diretamente, excluindo etapas intermediárias redundantes e demoradas. A abordagem dos autores é mais simples que outras da literatura, que produz detecção de texto rápida e precisa em cenas naturais. O método prevê diretamente palavras ou linhas de texto de orientações arbitrárias e formas quadrilaterais em imagens completas.

O trabalho de [Scheidl et al. 2018], propõe um sistema HTR (*Handwritten Text Recognition* - Reconhecimento de texto manuscrito) baseado em RNA (*Artificial Neural Network* - Rede Neural Artificial). O processo consiste em uma operação sequencial, onde a imagem do texto entra em uma CNN (*Convolutional Neural Network* - Rede Neural Convolucional), que são treinadas para extrair recursos relevantes da imagem. Logo após, os dados da CNN são passados para uma RNN, com uma sequência de 256 recursos por intervalo de tempo, onde o RNN propaga informações relevantes por meio dessa sequência. O autor utilizou então uma LSTM (*Long Short Term Memory* - Memória de Curto Prazo Longa) devido sua capacidade de propagar informações por distâncias maiores e oferecer características de treinamento mais robustas. Por fim, os resultados da RNN são passados então para a CTC, onde no final é realizado uma decodificação utilizando uma LM (*Language Model* - Modelo de Linguagem) própria denominada *word beam search*, aumentando a acurácia final do processo.

[Parthiban et al. 2020] propõe uma metodologia a partir da análise de RNNs (*Recurrent Neural Networks* - Redes Neurais Recorrentes), que é utilizada para descobrir a disposição dos caracteres, introduzindo uma rede neural recorrente para reconhecer textos escritos à mão. Há, segundo o autor, vários OCRs efetivamente acessíveis para múltiplos idiomas, porém a maioria são efetivos apenas para texto formal, mas para textos cursivos são incomuns, mostrando baixa precisão quando testados em textos do tipo *off-line*. O autor propõe uma arquitetura simples contendo camadas de convolução ligadas a camadas de recorrência, onde ao final é atribuído no processo uma camada CTC (*Connectionist Temporal Classification* - Classificação Temporal Conexionista).

3. Metodologia

Para realização do trabalho, foi planejada a implementação de algumas etapas, demonstradas no diagrama da Figura 1.

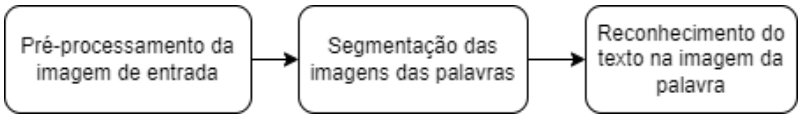


Figura 1. Diagrama demonstrando o processo de reconhecimento de palavras.

No pré-processamento, é realizado o *upscaling* (aumento de resolução) da imagem de entrada, para melhorar a extração de características das palavras pelo reconhecimento do texto na imagem da palavra. Na Figura 2 é demonstrado o resultado da utilização do algoritmo FSRCNN[2], realizando aumento de 4x na resolução da imagem de entrada.

[2]https://docs.opencv.org/4.x/d5/d29/tutorial_dnn_superres_upscale_image_single.html

Figura 2. Aplicação do *upscaling* de 4x.

Logo em seguida é aplicado o algoritmo *deskew* (endireitamento), que aplica dilatações e detecção de borda para então calcular o ângulo de inclinação da imagem. É possível visualizar o resultado na Figura 3.

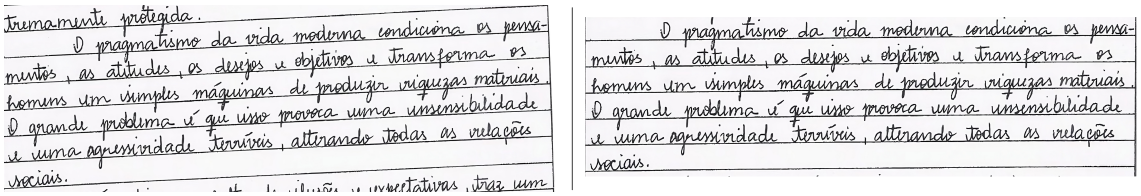


Figura 3. Aplicação de *deskew* na imagem de entrada.

Na segmentação das palavras foi usado então um modelo proposto por [Zhou et al. 2017] e [Axler and Wolf 2018], que classifica cada *pixel* como palavra (parte interna ou envolvente) ou *pixel* de fundo. Para cada *pixel* da classe de palavra interna, é prevista uma AABB (*Axis Aligned Bounding Box* - Caixa Delimitadora de Eixo Alinhado) em torno da palavra. Ao final é feito um agrupamento aos AABB previstos. O modelo é treinado no conjunto de dados *IAM Dataset* [Marti and Bunke 2002], onde é demonstrado sua execução na Figura 4.

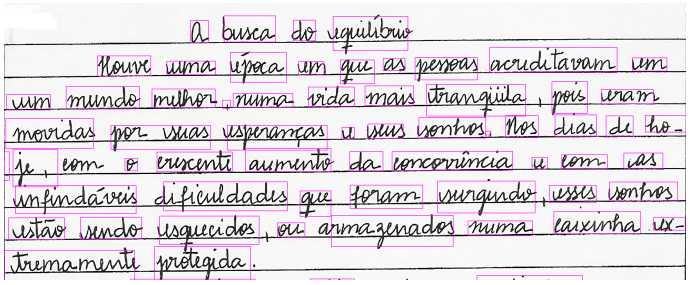


Figura 4. Modelo de detecção de palavras.

Para a remoção das linhas horizontais foram implementadas então técnicas de processamento de imagem, utilizando detecção de bordas, ajustando com dilatações e erosões para então detectar contornos e ao final remover linhas horizontais. Na Figura 5 é possível visualizar o resultado.

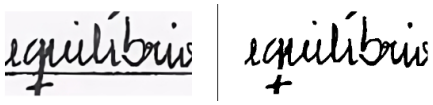


Figura 5. Remoção de linhas horizontais.

Já no reconhecimento do texto nas imagens das palavras, foram realizados vários testes na arquitetura, iniciando a partir do modelo proposto por [Scheidl et al. 2018], que

previa o reconhecimento de linhas de palavras. O modelo proposto neste trabalho analisa imagens de palavras isoladas, para tentar otimizar os resultados e diminuir a complexidade do modelo, com uma entrada de 128x32 *pixels*. Na Figura 6 pode-se observar a arquitetura proposta.

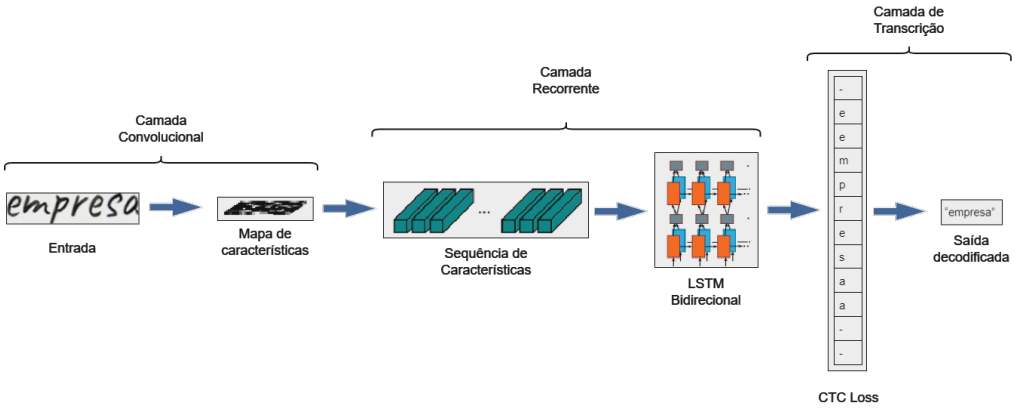


Figura 6. Arquitetura proposta para a tarefa de reconhecimento de caracteres.

A arquitetura em questão é denominada pela literatura de CRNN (*Convolutional Recurrent Neural Network* - Rede Neural Recorrente Convolucional), que consiste na concatenação de uma rede convolucional a uma rede recorrente. Ao final, a rede passa pela camada de transcrição, onde há o CTC, que gera pontuações de caracteres para cada elemento da sequência, que é representado por uma matriz. Uma camada CTC será responsável por propagar a atualização da rede e também será usada para a inferência no momento da decodificação do texto final.

Usar o CTC é interessante pois não seria necessário anotar a posição exata dos caracteres nas imagens de entrada. O CTC guia o treinamento usando a matriz da saída da RNN e o texto *ground truth* (palavra real que corresponde ao texto real na imagem), tenta todos os alinhamentos possíveis do *ground truth* na imagem e obtém a soma de todas as pontuações. Dessa forma, a pontuação de um *ground truth* é alta se a soma das pontuações de alinhamento tiver um valor alto.

4. Experimentos

Para os experimentos, foram categorizadas as imagens de 5 exemplos de redações pela qualidade da escrita e pela qualidade da imagem, sendo classificadas em: ótimo, bom, médio, ruim e péssimo. A Figura 7 mostra os resultados obtidos com o reconhecimento final do modelo proposto para cada uma das redações.

Inicialmente foi usado apenas o *IAM Dataset* para o treinamento e validação em textos reais. Porém verificou-se que o mesmo era pequeno para que a arquitetura generalizasse características da escrita do português brasileiro, que consiste de letras sobrepostas e ligaduras entre elas, além de acentos gráficos e pontuações.

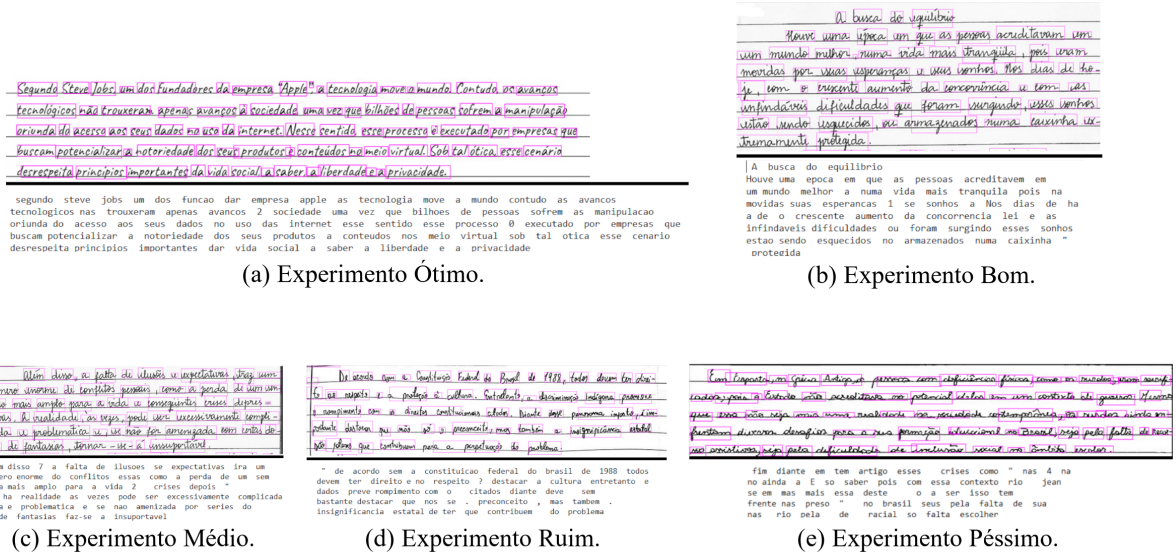
Assim, foi necessário construir um *dataset* próprio, nomeado de *HCAO Dataset*[3]. Esse *dataset* foi construido com 1432 palavras do português brasileiro, com 108 fontes

[3]*Handwritten Calligraphy with Accents and Overlays* - Caligrafia Manuscrita com Acentos e Sobreposições

que possuem caracteristicas de sobreposição de caracteres. Para cada palavra foi criado variações utilizando as fontes, resultando em 151500 imagens, com 108 fontes cursivas diferentes. O *IAM Dataset* possui 115320 imagens de palavras isoladas, com 657 fontes diferentes. Verificou-se que rodar treinos com os *datasets* separadamente não gerava bons resultados. Assim, no experimento, foi utilizada a junção de ambos os *datasets*, para melhorar a generalização do modelo.

Também foram avaliados dois decodificadores: *Beam Search* e *Lexicon Search*. O segundo apresentou melhorias significativas em comparação ao primeiro, que utiliza apenas a decodificação resultante da própria arquitetura. Porém para que o *Lexicon Search* apresente melhores resultados, o mesmo depende de um grande léxico[4]. Esse aumento acarreta em perda de desempenho na decodificação, pois quanto maior for a árvore do léxico mais demorado se torna o cálculo de aproximação dos caracteres com as inúmeras palavras disponíveis. No experimento, foi utilizado apenas o *lexicon search*, que apresentou melhores resultados.

Para o experimento, foi construído um dicionário com apenas 1.000 palavras mais usadas do português brasileiro para utilizar no decodificador. Na prática, os testes com textos mais diversificados no léxico, o decodificador não teve bons resultados devido ao tamanho reduzido de 1.000 palavras. Com um léxico acima de 10.000 palavras a decodificação passou a ser significativamente demorada.



(a) Experimento Ótimo. (b) Experimento Bom.

(c) Experimento Médio. (d) Experimento Ruim. (e) Experimento Péssimo.

Figura 7. Experimentos realizados com redações reais.

5. Resultados

A Tabela 1 mostra os resultados de um estudo comparativo com trabalhos da literatura utilizando o *IAM Dataset*, onde foi feito uma comparação das taxas CER e WER obtidas utilizando apenas esse *dataset*. Neste trabalho realizou-se uma média de três experimentos para calcular os dados informados, com uma divisão 95/5, ou seja, 95% dos dados foram usados para treinamento e 5% foram para validação. Não foi possível implementar os modelos da literatura utilizados no comparativo, então foi usado o mesmo *dataset*

[4]Léxico é o conjunto de palavras existente em um determinado idioma.

para termos de comparação e a mesma divisão dos dados treino e validação, analisando os resultados fornecidos pelos autores.

CER (*Character Error Rate* - Taxa de Erro de Caractere) é baseado no conceito da distância de *levenshtein*, onde é contado o número mínimo de operações por caracteres necessários para transformar o (*ground truth*) na palavra da saída da arquitetura. Esta taxa é calculada pela seguinte equação 1:

$$CER = (S + D + I)/N \tag{1}$$

onde S é o número de substituições, D é o número de remoções, I é o número de Inserções e N o número de caracteres do texto de referência (*ground truth*).

Já o WER (*Word Error Rate* - Taxa de Erro de Palavra) é aplicável na transcrição de parágrafos e frases de palavras com significado. Sua equação é idêntica ao CER, mudando apenas que o WER opera no nível da palavra, ou seja, substituições, remoções e inserções são feitas baseadas na palavra inteira, ao invés de caracteres por caracteres.

Tabela 1. Comparativo das taxas CER e WER com resultados da literatura.

Autor	CER(%)	WER(%)
Scheidl, Harald (2018)	4.86	10.15
Flor (2020)	8.58	27.90
Puigcerver	9.39	29.34
Bluche et al.	14.30	41.17
Este trabalho	9.10	27.81

Adicionalmente, realizou-se os cálculos de CER e WER para o nosso modelo treinado com a mistura do *IAM Dataset* e o *HCAO Dataset*, onde foi obtida uma melhoria moderada, pois embora haja uma visível melhoria nas taxas, em experimentos com palavras com fontes mais complexas, os resultados tendem a ser ruins devido ao enviesamento dos dados oriundos do *dataset*. Na Tabela 2 é apresentado os resultados.

Tabela 2. Resultado da junção do IAM Dataset com HCAO Dataset.

Autor	CER(%)	WER(%)
Este trabalho	3.74	10.42

Os trabalhos da literatura possuem certas variações da arquitetura, incluindo propostas diferentes na decodificação das palavras, onde são experimentados desde decodificação simples pela saída bruta da arquitetura pelo CTC até modelos de linguagem treinados. Porém, todas as arquiteturas são do tipo CRNN.

6. Conclusão

Este trabalho propôs uma metodologia para reconhecer textos manuscritos em imagens de redações, escritas por pessoas. A proposta é facilitar o envio da redação para sistemas de correção automática, auxiliando o profissional em suas atividades de correção, além de incentivar o candidato a escrever a redação em papel. Ao final, é possível a construção de *datasets* de textos para utilização em modelos de NLP para realização de correção automática de texto, pois com essa metodologia proposta, é possível a extração de textos em imagens de redações de sistemas de correção manual, já que muitos professores e escolas guardam redações de seus alunos na forma de imagens.

Esta metodologia apresentou resultados moderados para um escopo específico, porém é possível obter melhores resultados, realizando algumas mudanças na arquitetura proposta, onde ao invés de usar uma CNN para detectar características da imagem de entrada poderia ser utilizado curvas de *bézier* para extração direta das curvas da palavra e inserção na LSTM, pois tal técnica possibilita a obtenção de pontos de curva da escrita sem precisar treinar uma rede neural convolucional para extrair caracteristicas sobre as curvas, otimizando então os resultados.

Outra melhoria seria a troca da LSTM por uma arquitetura *Transformer*, para realizar o processamento dos dados de entrada utilizando o conceito de atenção, proposta por essa arquitetura. Em teoria seria possível obter uma melhoria na saída da arquitetura desenvolvida, pois o *Transformer* calcularia a probabilidade da sequência dos caracteres da palavra baseado em contexto de palavras reais, melhorando a utilização do decodificador na saída da arquitetura.

Referências

Axler, G. and Wolf, L. (2018). Toward a dataset-agnostic word segmentation method. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 2635–2639. IEEE.

Barros, S. C. B. d. (2019). Estudo do desempenho de candidatos à ufrn na prova de redação do enem no período de 2013 a 2016. Master's thesis, Brasil.

INEP, E. (2021). Painéis enem. *https://app.powerbi.com/view?r= eyJrIjoiYTdlOGQ3ZTgtMzc1Ny00ZDFkLTk4NjQtZDBkNTUyNjVhNmQ1Iiwid CI6IjI2ZjczODk3LWM4YWMtNGIxZS05NzhmLWVhNGMwNzc0MzRiZiJ9*, 1:1.

Marinho, J. C., Anchiêta, R. T., and Moura, R. S. (2021). Essay-br: a brazilian corpus of essays. *arXiv preprint arXiv:2105.09081*.

Marinho, J. C., Cordeiro, F., Anchiêta, R. T., and Moura, R. S. (2022). Automated essay scoring: An approach based on enem competencies. In *Anais do XIX Encontro Nacional de Inteligência Artificial e Computacional*, pages 49–60. SBC.

Marti, U.-V. and Bunke, H. (2002). The iam-database: an english sentence database for offline handwriting recognition. *International Journal on Document Analysis and Recognition*, 5:39–46.

Parthiban, R., Ezhilarasi, R., and Saravanan, D. (2020). Optical character recognition for english handwritten text using recurrent neural network. In *2020 International Conference on System, Computation, Automation and Networking (ICSCAN)*, pages 1–5. IEEE.

Scheidl, H., Fiel, S., and Sablatnig, R. (2018). Word beam search: A connectionist temporal classification decoding algorithm. In *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 253–258. IEEE.

Zhou, X., Yao, C., Wen, H., Wang, Y., Zhou, S., He, W., and Liang, J. (2017). East: an efficient and accurate scene text detector. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 5551–5560.