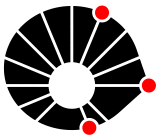

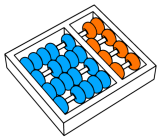

Otimização para Detecção de Texto Multilíngue Usando Redes Neurais Convolucionais

J. S. J. Conceição *R. S. Torres* *H. Pedrini*

Relatório Técnico - IC-PFG-22-19
Projeto Final de Graduação
2022 - Julho

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE COMPUTAÇÃO

Otimização para Detecção de Texto Multilíngue Usando Redes Neurais Convolucionais

Jhonatas Santos de Jesus Conceição* Ricardo da Silva Torres† Helio Pedrini‡

Julho de 2022

Resumo

Detecção de texto em cena tem recebido grande importância nos últimos anos. Os desafios desta tarefa consistem em conceber detectores capazes de lidar com uma ampla gama de variabilidade, tais como tamanho da fonte, estilo da fonte, cor, fundo complexo, entre outros fatores. Quando tratamos de textos multilíngue, as atuais propostas de detecção possuem dificuldade em detectar as diferentes línguas com o mesmo desempenho entre elas. Este trabalho apresenta uma técnica para otimizar as detecções individuais de línguas. Primeiro fazemos uma comparação entre duas estratégias de construção de modelos, utilizando redes neurais convolucionais, para detectar elementos textuais multilíngue em imagens: (i) modelo de detecção construído em um cenário de treinamento multilíngue e (ii) modelo de detecção construído em um cenário de treinamento específico de linguagem. A partir desta comparação, propomos um algoritmo de fusão dos treinamentos realizados com língua específica para podermos avaliar nossa hipótese no contexto de teste com todas as línguas. Os experimentos projetados neste trabalho indicam que o modelo específico de idioma supera o modelo de detecção treinado em um cenário multilíngue. Com o algoritmo de fusão dos modelos, obtivemos uma melhoria final de 28,21% e 11,80%, em termos de precisão e medida-F1, respectivamente.

1 Introdução

Informação textual embutidas em imagens ou vídeos provê valiosas fontes de informações para diversas aplicações de recuperação de dados multimídia baseada em conteúdo, as quais se utilizam de tais informações textuais com o objetivo de aumentar a precisão na busca por conteúdo relacionado a um determinado evento de interesse [1, 2], no entendimento semântico de uma cena [3], navegação autônoma de robôs [4], rastreamento de texto em estradas [5], veículos autônomos [6], entre outros.

Diferentemente dos problemas de reconhecimento de placas veiculares e do reconhecimento de textos em documentos bem formatados, a tarefa de detecção e reconhecimento de textos em cenas se apresenta como um cenário mais desafiador devido aos diferentes fatores

*Instituto de Computação, Universidade Estadual de Campinas, Campinas, SP, Brasil, 13083-852.
†Norwegian University of Science and Technology, Høgskoleringen 1, 7034 Trondheim, Noruega.
‡Instituto de Computação, Universidade Estadual de Campinas, Campinas, SP, Brasil, 13083-852.

de variabilidade inerentes ao problema tais como a orientação dos textos, diferentes tamanhos e estilos das fontes, planos de fundo, texturas, cores e o contexto da linguagem. Estes e outros fatores de variabilidade adicionam grandes desafios e problemáticas que limitam o uso dos métodos tradicionais de reconhecimento de caractere óptico (*do inglês, Optical Character Recognition* – OCR).

Podemos categorizar as informações textuais encontradas em imagens em textos gráficos e textos da cena [7]. Em suma, textos gráficos podem ser entendidos como elementos textuais impressos digitalmente sobre um elemento gráfico tais como os textos encontrados em legendas, anotações feitas em imagens em vídeos, e imagens provenientes da web (por exemplo, *Facebook*[1], *YouTube*[2]) e de aplicativos de comunicação (por exemplo, *WhatsApp*[3], *Outlook*[4]). Já os textos da cena podem ser entendidos como os elementos textuais capturados nativamente no ambiente, durante o processo de aquisição da cena. Exemplos de textos de cena incluem os textos em sinalização, cartazes e painéis publicitários, pacotes de encomendas, placas escritas a mão, letreiros digitais, elementos textuais em roupas, entre outros.

De acordo com Ye e Doermann [7], o problema de reconhecimento em texto em cenas naturais pode ser dividido nos seguintes subproblemas: (1) localização de regiões candidatas a conterem uma informação textual; (2) procedimentos de verificação para determinar se uma região possui alguma informação textual; (3) extração da informação textual; e (4) reconhecimento dos caracteres que compõe o texto extraído. Embora seja possível identificar problemáticas inerente a cada subproblema, recentes trabalhos sugerem que a integração, usando esquemas de retroação (*feedback*), das tarefas de detecção e reconhecimento de textos são importantes para se obter soluções mais robustas.

Neste trabalho, lidamos com o problema de localização de textos em regiões candidatas e classificação das mesmas. Por isso, tem como objetivo investigar o uso de tecnologias e técnicas de auxílio em detecção de textos para cenários multilíngue, com intuito de obter melhores resultados de precisão, revocação e medida-F1 para cada língua individualmente.

Nas atuais abordagens de detecção de texto, encontramos diversos métodos que trabalham com esta tarefa usando técnicas fim-a-fim [5, 8], nas quais compreende diferentes línguas para uma mesma arquitetura. Em muitos casos, apesar de obter resultados satisfatórios, estas técnicas não conseguem generalizar suficientemente bem e igualmente para todas as línguas envolvidas no processo de treinamento.

Neste trabalho, partimos da hipótese de que, ao realizarmos o treinamento de línguas especificas separadamente, os resultados do fluxo escolhido, no nosso caso a rede PixelLink [9], tendem a melhorar. Tendo em vista a resposta desta hipótese, também criamos um algoritmo para reagrupar os treinamentos realizados individualmente, pois, apesar de buscarmos aumentar a precisão dos resultados individuais de cada língua, em cenários de testes, ainda precisamos lidar com a incerteza da língua da região detectada e, portanto, nossa solução final precisa ser integrada.

Este trabalho está organizado como segue. A Seção 2 descreve brevemente os recentes

[1] https://www.facebook.com/
[2] https://www.youtube.com/
[3] https://www.whatsapp.com/
[4] https://outlook.live.com

trabalhos desta área. A Seção 3 apresenta os conjuntos de dados, as métricas e as arquiteturas utilizados nos experimentos deste trabalho. A Seção 4 descreve a metodologia experimental para alcançar os objetivos propostos. A Seção 5 reporta e analisa os resultados obtidos em cada experimento realizado. A Seção 6 apresenta algumas considerações finais.

2 Trabalhos Relacionados

Esta seção descreve brevemente alguns trabalhos relacionados ao tema sob investigação neste trabalho.

2.1 Localização de Textos

A localização de textos em uma cena consiste na identificação de regiões presentes em imagens e vídeos como possível local contendo os textos. Este problema tem recebido significativa atenção nos últimos anos e vários métodos baseados em aprendizado profundo [10–13] foram relatados na literatura.

Revisões abrangentes e análises detalhadas podem ser encontradas em alguns levantamentos de pesquisa (*surveys*) [14]. Os primeiros métodos baseados em redes neurais profundas podem ser consultados em [15, 16] e, geralmente, consistem em estágios, como agregação de candidatos, partição de palavras e remoção de falsos positivos filtrados em etapas de pós-processamento. Nesta subseção, apresentamos alguns trabalhos deste campo de pesquisa a partir de dois grupos de métodos: algoritmos baseados em regressão e algoritmos baseados em segmentação.

2.1.1 Algoritmos Baseados em Regressão

Muitos trabalhos consideram palavras ou linhas de texto como objetos e adaptam abordagens de detecção geral de objetos, por exemplo, Faster R-CNN [17], SSD [18] e YOLO [19] na detecção de textos. Inspirados na Faster R-CNN, muitos algoritmos baseados em regressão usam vários *kernels* de convolução de diferentes escalas para gerar um grande número de *bounding boxes* de âncora no mapa de características e, em seguida, usam um algoritmo simples de supressão não máxima [20] para filtrar mais caixas de texto. TextBoxes [21] modificam a SSD usando *kernels* convolucionais irregulares e longas âncoras de acordo com a característica do texto da cena, mas só é capaz de cobrir áreas de texto horizontais.

O método TextBoxes++ [11] foi criado para melhorar TextBoxes permitindo que âncoras horizontais possam abranger quadriláteros mais gerais em textos orientados. Também propõe uma supressão não máxima em cascata, que é eficiente para *bounding boxes* rotacionados.

A R-YOLO [22] usou a YOLO como base para criar um modelo de rede neural convolucional (CNN) para detectar textos orientados arbitrariamente em cenas de imagens naturais. Para isso, são usadas informações de angulação na geração das âncoras.

ABCNet [23] usou uma curva de Bèzier para ajustar regiões de texto e a detecção de texto de forma arbitrária melhorou significativamente em termos de precisão e eficiência.

A ContourNet [24] contém um módulo Adaptive RPN para gerar propostas de texto focando apenas nos valores de *Intersection over Union* (IoU) entre *bounding boxes* preditos e *ground truth*. Além disso, um novo módulo de reconhecimento de textura ortogonal local (LOTM) que modela as informações de textura local foi proposto para evitar detecções de falsos positivos.

Ma et al. [25] dedicaram-se a resolver o problema de texto multidirecional e propuseram um algoritmo RRPN para gerar *bounding boxes* rotativos, introduziram informações de ângulo e propuseram uma camada de *pooling* de uma Região de Interesse Rotacionada (RRoI) para combinar orientações arbitrárias. Essa proposta foi projetada no mapa de características de um classificador de região de texto. O método melhorou a eficiência computacional de detecção de texto orientada de forma arbitrária.

2.1.2 Algoritmos Baseados em Segmentação

Algoritmos baseados em segmentação usam redes totalmente convolucionais (FCN) para segmentar regiões de texto em segundo plano. Para tal, inicialmente é predito um mapa de pontuação pela rede e, em seguida, os pixels de texto são agrupados, a máscara de texto é gerada e os pixels de texto são mesclados.

Neste contexto, é comum os textos possuírem uma forma arbitrária. Para lidar com este problema, foi proposto o algoritmo Mask TextSpotter [26], que é baseado no *framework* Mask R-CNN, e que segmenta cada caractere individualmente. A detecção e o reconhecimento precisos de texto foram adquiridos por meio de segmentação semântica. Ele tornou-se um método adequado para o tratamento de instâncias de texto em formas irregulares como texto curvo.

A rede PSENet [27] lida com o problema de instâncias de texto adjacentes que são unidas na detecção. Esta abordagem facilita o uso de métodos baseados em segmentação para detectar texto com formato arbitrário. A rede PANet [28] propõe um pós-processamento de agregação de pixels de texto a fim de reduzir o tempo de inferência do detector.

DBNet [29] contém um módulo Binarização diferenciável (DB) para uma rede de segmentação. Também propuseram um modelo de Fusão de Escala Adaptativa(ASF) para melhorar a robustez da escala.

O método para detecção de texto multilíngue usado neste trabalho, a PixelLink [9], também é baseado em segmentação semântica das imagens. Mais detalhes serão apresentados na próxima seção.

3 Materiais

Esta seção apresenta os conjuntos de dados, as métricas e as arquiteturas utilizadas nos experimentos deste trabalho para validar a hipótese planejada e melhorar os resultados de desempenho de métodos de localização de texto em um cenário multilíngue.

3.1 Conjuntos de Dados

Neste trabalho, usamos quatro conjuntos de dados amplamente empregados para projetar e avaliar métodos de localização e reconhecimento de textos, os conjuntos de dados ICDAR 2015 [30], MLT 2017 [31], MLT 2019 [32] e COCO-Text [33], que são descritos nesta seção. Estes conjuntos de dados estão relacionados à Competição de Leitura Robusta (*Robust Reading Competition*), que se refere a técnicas e metodologias que não envolvem apenas documentos de texto em papel digitalizado, mas também imagens e vídeos com texto.

Por isso, as técnicas da literatura para detecção e reconhecimento de texto são publicadas nesta competição. Uma de suas modalidades abrange o foco principal deste trabalho, ou seja, a detecção e o reconhecimento de texto em cenário multilíngue(*multi-lingual scene text detection and recognition*).

ICDAR 2015: Este conjunto de dados [30] contém 1500 imagens, sendo 1000 imagens de treinamento e 500 imagens de teste. As imagens foram capturadas pelos óculos do Google e contêm textos com diferentes orientações, desfocados ou com baixa resolução. As anotações foram construídas em termos de retângulos envolventes (*bounding boxes*) de palavras. A Figura 1 ilustra exemplos desse conjunto de dados.

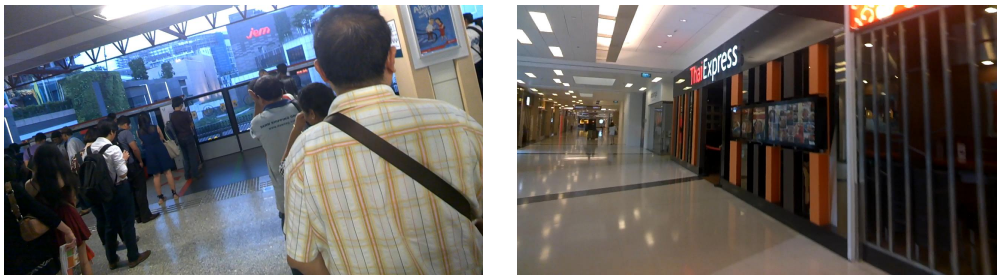

Figura 1: Exemplos de imagens do conjunto de dados ICDAR 2015.

MLT 2017: Este conjunto de dados [31] compreende 18000 imagens contendo texto de nove idiomas, 2000 imagens por idioma, incluindo árabe, bangla, chinês, inglês, francês, alemão, italiano, japonês e coreano. No total, esse conjunto de dados contém 9000 imagens de treinamento e 9000 imagens de teste. A Figura 2 ilustra exemplos desse conjunto de dados.

MLT 2019: Este conjunto de dados [34] contém 10000 imagens de treinamento e 10000 imagens de teste contendo imagens de cena com texto em 10 idiomas, 1000 imagens por idioma, incluindo árabe, bangla, chinês, hindi, inglês, francês, alemão, italiano, japonês e coreano. A Figura 3 ilustra exemplos desse conjunto de dados.

COCO-Text: Este conjunto de dados [33] de grande escala é usado para detecção e reconhecimento de texto em imagens naturais. Neste trabalho, usamos a versão 1.4 do conjunto de dados COCO-Text com 63686 imagens, sendo 43686 para treinamento, 10000

Figura 2: Exemplos de imagens do conjunto de dados MLT 2017.

Figura 3: Exemplos de imagens do conjunto de dados MLT 2019.

para validação, e 10000 para teste (sem anotações públicas neste ultimo). A Figura 4 ilustra exemplos desse conjunto de dados.

Figura 4: Exemplos de imagens do conjunto de dados COCO-Text.

3.2 Métricas de Avaliação

Para avaliar os resultados, adotamos as medidas de Precisão, Revocação e medida-F1, amplamente utilizadas na literatura. Para calcular essas métricas, primeiramente forneceremos algumas definições básicas:

- Verdadeiro Positivo (VP): detecção correta feita pelo modelo.
- Falso Positivo (FP): detecção incorreta feita pelo modelo.
- Falso Negativo (FN): *ground truth* não detectado pelo modelo.

- Verdadeiro Negativo (VN): região de fundo não detectada corretamente pelo modelo. Esta métrica não é usada propriamente em detecções de objetos, pois não há anotações deste tipo de região.

A precisão, a revocação e a medida-F1 podem ser calculadas usando as fórmulas fornecidas nas seguintes equações:

$$\text{Precisão} = \frac{\text{VP}}{(\text{VP} + \text{FP})} = \frac{\text{VP}}{(\text{todas as detecções})} \tag{1}$$

$$\text{Revocação} = \frac{\text{VP}}{(\text{VP} + \text{FN})} = \frac{\text{VP}}{(\text{todos os } \textit{ground truth})} \tag{2}$$

$$\text{Medida-F1} = \frac{2\,(\text{Precisão} \times \text{Revocação})}{(\text{Precisão} + \text{Revocação})} \tag{3}$$

No problema de detecção de texto, a precisão mede a fração de *bounding boxes* corretos sobre todas os *bounding boxes* detectadas com o método. Com isso, esta métrica consegue expressar qual o grau de exatidão do modelo em identificar objetos relevantes. A revocação mede a fração de *bounding boxes* corretos detectados sobre todas os *bounding boxes* presentes no *ground truth*. Com isso, esta métrica mede a capacidade do modelo de detectar todos os *ground truths*. Finalmente, a medida-F1 é dada pela média harmônica entre precisão e revocação.

Outra métrica utilizada em um dos experimentos realizados neste trabalho, não como avaliação mas como seleção, é a Intersecção sobre a União (*Intersection over Union* - IoU). Em detecção de objetos, esta métrica avalia o grau de sobreposição entre o *ground truth* e a predição realizada pelo modelo. Estas predições e o *ground truth* podem ter qualquer formato, entretanto, em nosso caso de uso, lidaremos apenas com retângulos. A IoU pode ser calculada com a fórmula expressa na Equação 4.

$$\text{IoU} = \frac{\text{área}\ (\textit{ground truth} \cap \text{predição})}{\text{área}(\textit{ground truth} \cup \text{predição})} \tag{4}$$

A IoU varia entre 0 e 1, em que 0 não mostra nenhuma sobreposição e 1 significa sobreposição perfeita entre *ground truth* e predição. Esta métrica é útil para que possamos decidir, a partir de um limiar, qual o critério de aceitação dos *bounding boxes* preditos.

3.3 Arquiteturas

Esta seção apresenta a explicação dos métodos da literatura usados como base nos experimentos realizados. A seguir, descrevemos o método PixelLink [9], que é uma rede neural convolucional projetada para classificar os pixels de uma imagem como texto/não-texto, e também para prever os links entre eles e chegar a uma localização de texto baseada em palavras. Em sequência, descrevemos a Faster R-CNN [17], que faz parte de uma série de artigos importantes sobre detecção de objetos, sendo antecedida pelos métodos R-CNN [35] e Fast R-CNN [36]. Entre os incrementos das versões, a evolução se deu em termos de

eficiência computacional, desempenho e redução no tempo de teste. Geralmente essas redes consistem em um algoritmo para gerar propostas de localizações de objetos na imagem, uma etapa de geração de características desses objetos usando CNN, uma etapa para classificar o objeto e uma camada de regressão para tornar os *bounding boxes* mais precisos.

3.4 PixelLink: Visão Geral

O método PixelLink aborda o problema de detecção de texto em uma cena com base na segmentação de instância. No problema de segmentação de instâncias, há duas tarefas principais envolvidas: (i) predição de categorias para pixels de uma imagem realizando uma rotulagem de pixel e (ii) diferenciação de objetos de uma mesma categoria (por exemplo, segmentar indivíduos em multidões, carros em trânsito intenso).

No contexto do problema de localização de texto, a PixelLink prevê pixels positivos, ou seja, pixels pertencentes a elementos textuais, e os une em instâncias de texto prevendo *links* positivos. Para vincular um pixel a outro, a PixelLink verifica seus vizinhos, considerando uma vizinhança de oito conexões, para verificar se existe algum vizinho rotulado como pixel positivo. Assim, pixels positivos são agrupados para formar componentes conexos, em que cada componente conexo representa uma instância de texto. Por fim, o método calcula um *bounding box* de área mínima de todas as instâncias de texto encontradas e remove todas os *bounding boxes* com um lado menor do que 10 pixels ou uma área menor do que 300 pixels. Na Figura 5, pode-se observar o fluxo geral de segmentação semântica da rede PixelLink.

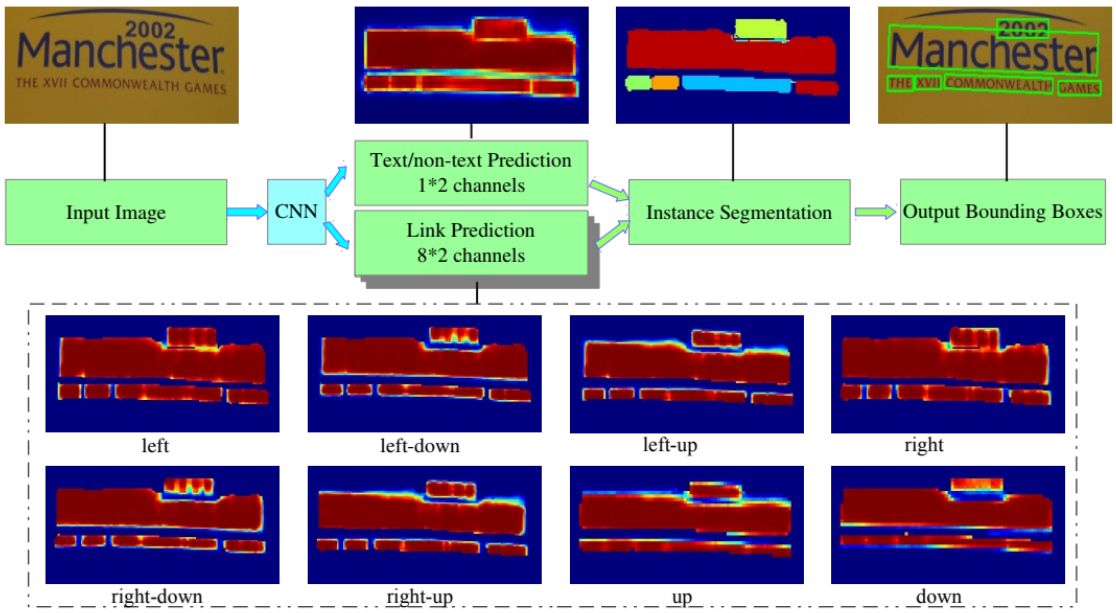

Figura 5: Representação do fluxo de segmentação semântica da rede PixelLink [9].

3.5 PixelLink: Funções de Perda

A PixelLink define três funções de perda para (i) examinar cada pixel individualmente, (ii) examinar os *links* previstos para os pixels de ligação da mesma instância e (iii) calcular o erro geral na fase de treinamento. A Equação 5 mostra a perda de treinamento, que consiste na soma ponderada das perdas de pixel (L_{pixel}) e de *link* (L_{link}):

$$L = \lambda L_{pixel} + L_{link} \tag{5}$$

A Equação 6 mostra a função de perda de pixels usada neste trabalho, em que r refere-se à razão positivo-negativo, S refere-se à área da instância e W é uma matriz de pesos para todos os pixels positivos que é usada para equilibrar a perda calculada sobre pequenas e grandes áreas (Equação 8), para todas as N instâncias. Finalmente, o L_{pixel_CE} é a matriz de perda de entropia cruzada calculada para as previsões de texto e não-texto:

$$L_{pixel} = \frac{1}{(1+r)S} + WL_{pixel\_CE} \tag{6}$$

$$w_i = \frac{B_i}{S_i} \tag{7}$$

$$B_i = \frac{S}{N} \quad S = \sum_{i}^{N} S_i \quad \forall i \in \{1, \cdots, N\} \tag{8}$$

Por sua vez, a perda de *link* é definida como a soma das perdas de *links* positivas e negativas, conforme mostrado nas equações a seguir:

$$L_{link} = \frac{L_{link\_pos}}{rsum(W_{pos\_link})} + \frac{L_{link\_neg}}{rsum(W_{neg\_link})} \tag{9}$$

$$L_{link\_pos} = W_{pos\_link} L_{link\_CE} \tag{10}$$

$$L_{link\_neg} = W_{neg\_link} L_{link\_CE} \tag{11}$$

$$W_{pos\_link}(i,j,k) = W(i,j) \times (Y_{link(i,j,k)==1)} \tag{12}$$

$$W_{neg\_link}(i,j,k) = W(i,j) \times (Y_{link(i,j,k)==0)} \tag{13}$$

em que k é o k-ésimo vizinho do pixel (i,j), $rsum$ é uma função de soma reduzida que calcula a soma de todos os elementos de um tensor, W é a matriz de pesos definida na Equação 6 e Y é a matriz de rótulo de ligações.

3.6 Faster R-CNN

O artigo da Faster R-CNN apresenta um fluxo de detecção que usa a rede de proposta de regiões (*Region Proposal Network* - RPN) como algoritmo para gerar regiões candidatas que, por fim, alimentam a rede Fast R-CNN como rede para detecção. Todo o sistema é uma rede única e unificada para detecção de objetos (Figura 6).

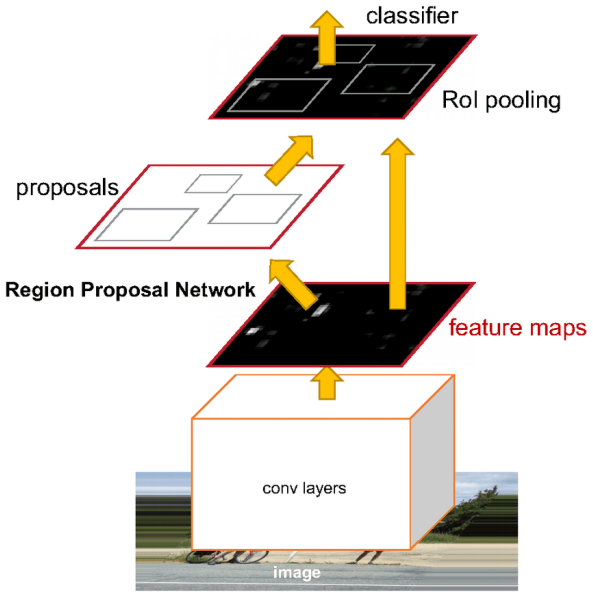

Figura 6: Representação da arquitetura da rede Faster R-CNN [17].

3.6.1 Rede de Proposta de Regiões

A rede de proposta de regiões (RPN) recebe como entrada uma imagem de dimensão 1000×600, alimentando uma rede base VGG-16 ou ZF-Net. Em ambas as redes, há uma redução de dimensionalidade das características de entrada. Para cada ponto no mapa de características de saída, a rede precisa identificar se um objeto está presente na imagem de entrada e estimar seu tamanho. Isso é feito colocando um conjunto de "âncoras" na imagem para cada local no mapa de características de saída da rede base. Essas âncoras indicam possíveis objetos em vários tamanhos e proporções naquele local. No artigo, as âncoras utilizadas possuem 3 escalas de área $128^2, 256^2, 512^2$ e 3 proporções de $1:1$, $1:2$ e $2:1$.

À medida que a rede se move através de cada pixel no mapa de características de saída, ela deve verificar se essas k âncoras contêm objetos para refinar as coordenadas e fornecer regiões de interesse. Como passo seguinte, esses mapas de características servem de entrada para duas camadas irmãs totalmente conectadas, sendo elas uma camada de regressão de *bounding box* e uma de classificação. A saída da classificação fornece a probabilidade de cada ponto no mapa de características da rede base contenha um objeto dentre todas as 9 âncoras naquele ponto. Já a regressão fornece coeficientes que são usados para melhorar as coordenadas das âncoras que contêm objetos.

Uma âncora é considerada uma amostra "positiva" se satisfizer uma das duas condições: a) a âncora tem a maior IoU (Intersecção sobre União) com um *bounding box* de *ground truth*; b) a âncora tem uma IoU maior de 0,7 com qualquer *bounding box* de *ground truth*. Um mesmo *bounding box* de *ground truth* pode fazer com que várias âncoras sejam atribuídas a rótulos positivos. Uma âncora é rotulada como "negativa" se sua IoU com todas as *bounding boxes* de *ground truth* for menor que 0,3.

A função de perda utilizada nesta arquitetura é definida na Equação 14.

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}(\sum_i L_{cls}(p_i, {p_i}^*)) + \frac{\lambda}{N_{reg}}(\sum_i {p_i}^* \times L_{reg}(t_i, {t_i}^*)) \tag{14}$$

em que
i = índice de uma âncora.
p_i = probabilidade predita da âncora i ser um objeto.
${p_i}^*$ = valor de *ground truth* das âncoras, 1 se a âncora é positivo e 0 caso contrário.
t_i = coordenadas das âncoras previstas.
t_i^* = coordenada de *ground truth* associada aos *bounding boxes*.
L_{cls} = perda do classificador.
L_{reg} = perda de regressão.
N_{cls} = parâmetro de normalização do tamanho do mini-lote.
N_{reg} = parâmetro de normalização da regressão.
$\lambda = 10$ para tornar o parâmetro de perda cls e reg com peso aproximado.

A perda de regressão $L_{reg}(t_i, {t_i}^*)$ é ativada somente se a âncora realmente contiver um objeto, ou seja, se o *ground truth* ${p_i}^*$ for 1. Para a regressão de *bounding boxes*, eles adotam as parametrizações das 4 coordenadas como mostra a Equação 15.

$$\begin{aligned} t_x &= \frac{(x - x_a)}{w_a}, \quad t_y = \frac{(y - y_a)}{h_a}, \quad t_w = \log\left(\frac{w}{w_a}\right), \quad t_h = \log\left(\frac{h}{w_a}\right) \\ t_x^* &= \frac{(x^* - x_a)}{w_a}, \quad t_y^* = \frac{(y^* - y_a)}{h_a}, \quad t_w^* = \log\left(\frac{w^*}{w_a}\right), \quad t_h^* = \log\left(\frac{h^*}{h_a}\right) \end{aligned} \tag{15}$$

em que x, y, w e h denotam as coordenadas do centro do *bounding boxes*, sua largura e altura. As variáveis x, x_a e x^* são para *bounding box* predito, a âncora e *ground truth*, respectivamente (da mesma forma para y, w, h).

3.7 Fast R-CNN

Na Fast R-CNN, a imagem de entrada é passada primeiramente por uma rede base para obter o mapa de características (dimensão: 60, 40, 512). Neste passo, também ocorre o compartilhamento de peso entre a rede base da RPN e a rede base do detector Fast R-CNN. Em seguida, as propostas de *bounding boxes* da RPN são usadas para agrupar características do mapa de características da rede base. Isso é feito pela camada de *pooling* da Região de interesse (RoI).

A camada de *pooling* da RoI, em essência, funciona tomando a região correspondente a um mapa de características da rede base, dividindo esta região em um número fixo de sub-janelas e executando um *pooling* de máximo nessas sub-janelas para fornecer uma saída de tamanho fixo.

Depois de geradas as saídas de tamanho fixo, os mapas de características passam por duas camadas totalmente conectadas, e, em seguida, alimentam um classificador e um regressor. Vale lembrar que essas ramificações de classificação e regressão são diferentes daquelas da RPN.

A camada de classificação possui n unidades para cada uma das tarefas de detecção, então os mapas de características passam por uma softmax para obter a probabilidade de uma região proposta pertencer a cada classe. Os coeficientes da camada de regressão, são usados para melhorar os *bounding boxes* preditos. Este regressor é específico para cada classe e, dessa forma, todas as classes possuem regressores individuais com 4 parâmetros, cada um correspondendo a unidades de saída $n * 4$ na camada de regressão.

4 Metodologia

Esta seção apresenta os protocolos e a descrição dos experimentos projetados para validar nossa hipótese. Em uma primeira parte, elaboramos os experimentos para validar a hipótese de que modelos específicos funcionam melhor do que modelos gerais em um cenário de detecção de texto multilíngue. Esta resposta já foi obtida em [37], entretanto, considerando apenas o cenário da língua árabe. Neste artigo, o treinamento realizado especificamente se mostrou melhor do que um treinamento geral. Porém, realizaremos a avaliação em várias outras línguas para validar esta hipótese.

Em sequência, elaboramos um algoritmo com intuito de realizar a fusão dos modelos específicos, visto que, independente das nossas conclusões para a primeira parte, ainda é necessário que a resposta final esteja agrupada em uma única solução. Um fluxo geral para nossas experimentações é mostrado na Figura 7.

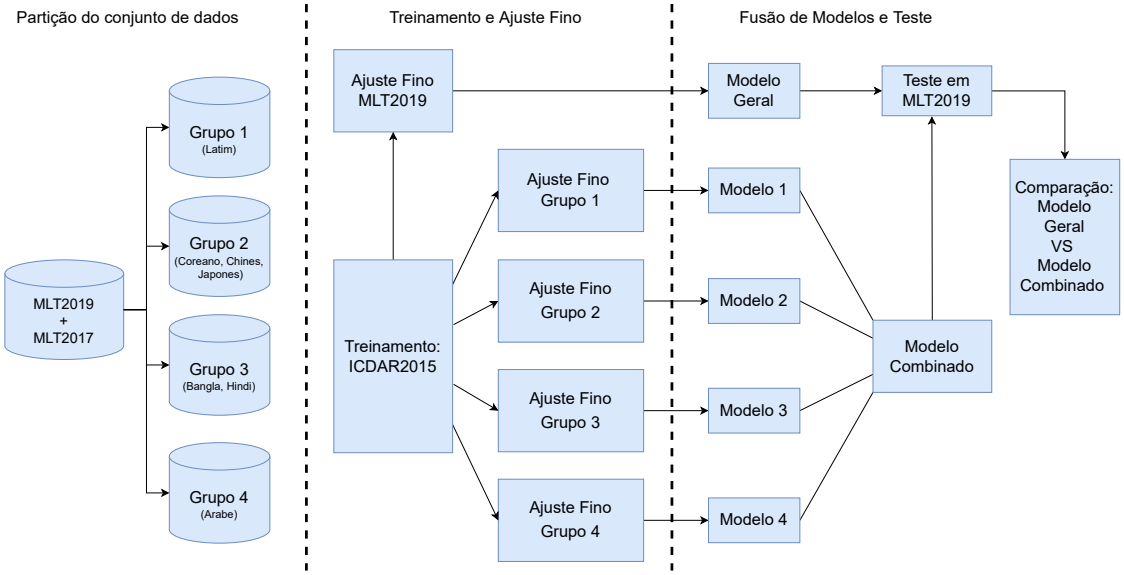

Figura 7: Resumo dos experimentos reportados na metodologia.

4.1 Modelo Geral × Modelo Específico

Esta subseção descreve o modelo geral e o modelo específico no cenário multilíngue explorado neste trabalho.

4.1.1 Experimento 1: Detecção de Texto Considerando um Treinamento Multilíngue

Este experimento visa verificar os resultados de desempenho da rede PixelLink em um cenário multilíngue. Inicialmente, realizamos um pré-treinamento da rede, considerando o conjunto de dados ICDAR 2015, com 110 épocas. Em seguida, realizamos um novo treinamento (*fine tuning*) usando o conjunto de treinamento da competição MLT 2019, também com 110 épocas. Então, testamos o modelo no conjunto de validação oficial da competição.

4.1.2 Experimento 2: Localização de Texto via Modelo Específico de Idioma

Os resultados obtidos no experimento anterior nos motivaram a investigar um esquema de treinamento considerando uma língua específica. Nossa hipótese é que a rede PixelLink não foi capaz de codificar adequadamente a especificidade dos idiomas fornecidos no conjunto de dados MLT 2019, como símbolos, *links* entre caracteres e espaçamento entre palavras e caracteres.

Para verificar nossa hipótese, dividimos os idiomas presentes nos conjuntos de dados em quatro grupos com semelhanças morfológicas. Desta forma, obtivemos os grupos:

- Grupo 1: latim;
- Grupo 2: chinês, japonês e coreano;
- Grupo 3: bangla e hindi;
- Grupo 4: árabe.

Neste experimento, utilizamos para treinamento os dados de treinamento da competição MLT2017 e os dados da competição MLT2019. Como teste, usamos o conjunto de validação da competição MLT2019. Assim como no experimento anterior, também realizamos um pré-treinamento no conjunto de dados ICDAR 2015.

4.1.3 Configuração Experimental

A fase de treinamento da rede PixelLink foi realizada utilizando imagens RGB de entrada com 512×512 pixels, taxa de aprendizado de 10^{-3} e um *batch size* de 8. Também usamos o método *Online Hard Example Mining* (OHEM) [38] para selecionar pixels negativos para ter uma proporção de pixels negativos-positivos de 3. Finalmente, definimos $\lambda = 2.0$, na Equação 5, a fim de dar explicitamente mais importância para a tarefa de rotulagem *pixel-wise*. A PixelLink foi implementado em Python usando o pacote TensorFlow.

Todos os experimentos foram conduzidos em um processador Intel Core i7-8700 @3.20GHz com 62GB de RAM e Nvidia RTX 2080 Ti 11GB executando um sistema operacional Linux.

4.2 Algoritmo para Fusão de Modelos Específicos de Idiomas

Nesta segunda parte da seção, mostraremos uma proposta de solução para o problema mencionado nos tópicos anteriores. Ela consiste na implementação de um algoritmo que viabiliza a fusão dos resultados dos modelos específicos em um resultado integrado, possibilitando, assim, a comparação com um modelo geral em um conjunto de teste com todos os idiomas da competição.

4.2.1 Visão Geral da Solução

Na Figura 8, elaboramos um resumo visual das etapas executadas para obter a fusão dos modelos específicos.

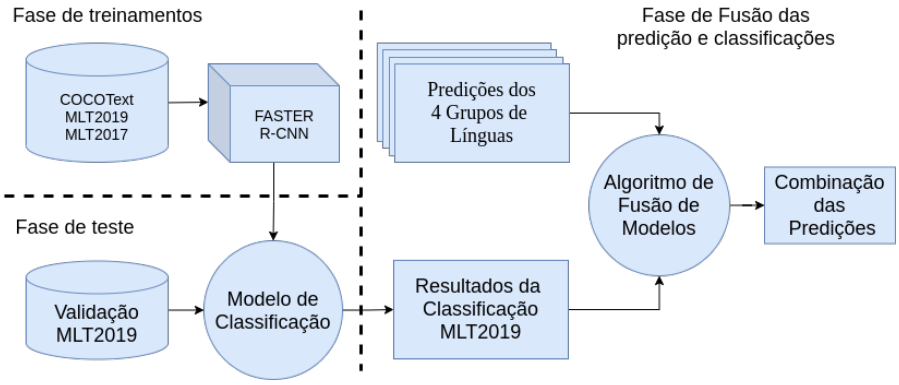

Figura 8: Visão geral do desenvolvimento da fusão de modelos específicos.

- **Fase de Treinamento:** nesta primeira etapa, selecionamos a base COCO-TEXT, para compor um pré-treinamento da rede Faster RCNN. Em seguida, fizemos o ajuste fino para o problema de classificação com 90% das bases MLT2017 e MLT2019. Com este treinamento, criamos um modelo para classificação de *bounding box* para as seguintes classes: Latin, cjk, bh, Arabic, symbols e ### (não identificado).

- **Fase de Teste:** na sequência, utilizamos o modelo de classificação no conjunto de validação oficial da competição MLT2019.

- **Fase de Fusão das Predições e Classificações:** Após a classificação, este resultado e o resultado obtido da predição de modelos específicos (Subseção 4.1.2), no mesmo conjunto de validação, serviram como entrada para nosso algoritmo de fusão, que será explicado em mais detalhes na próxima subseção. Com isso, o resultado deste passo corresponde à detecção final do conjunto de validação MLT2019 e, portanto, é comparável à detecção do modelo geral que realiza o treinamento e o teste em todo o conjunto de dados.

4.2.2 Algoritmos

Nesta seção, forneceremos mais detalhes dos algoritmos desenvolvidos para realizar a fusão das predições dos modelos específicos. Para este propósito, desenvolvemos duas funções: o Algoritmo 1: FUSÃO-DE-MODELOS e o Algoritmo 2: BUSCA-BBOX. A ideia geral da solução é, a partir do conjunto de imagens classificadas por nossa rede vista na seção anterior, encontrarmos o *bounding box* correspondente no conjunto das predições dos grupos específicos. Dessa forma, conhecendo-se a classe deste *bounding box*, conseguimos verificar entre as predições daquela classe se há uma correspondência de *bounding box* através do cálculo de IoU destas regiões.

O Algoritmo 1 descreve nossa solução principal. Nela, recebemos como entrada, os dados da classificação obtida por nossa rede na seção anterior, os quatro conjuntos de predições dos nossos modelos específicos. Este possuem as informações de *id* da imagem, as instâncias que correspondem aos *bounding boxes* da imagem, todos com uma determinada confiança vinculada. Também recebemos mais dois parâmetros de limiar: limiar de confiança, e limiar de IoU. Ambos servem como critério para filtrarmos os *boundings boxes* das predições e da classificação. Como retorno, esta função fornece o conjunto de *bounding boxes* selecionados de cada grupo e para cada imagem, ou seja, o conjunto contendo a fusão das predições dos modelos específicos.

No algoritmo, primeiramente definimos a variável a qual alocaremos a fusão (linha 2). Então percorremos todas as imagens presentes nos dados que foram classificados (linha 3). Para cada imagem, inicializamos a variável que conterá as instâncias com a fusão dos *bounding boxes*, consideramos o *id* da imagem a qual estamos analisando e armazenamos esta informação a uma variável que conterá as instâncias e o *id* da imagem após a fusão (linhas 4 – 7). Em sequência, visitamos todas as instâncias presentes em uma imagem (linha 8). Para cada uma, verificamos se a confiança do *bounding box* é menor do que o limiar de confiança. Caso seja, ignoramos esta instância (linhas 9–10). Caso contrário, consideramos a classificação da instância e, através da função do Algoritmo 2, buscamos nas predições destinadas a esta classe (conjunto predições 1: classificação latim, predição 2: chines, japonês e coreano, predição 3: bangla e hindi, predição 4: árabe). Para isso, também usamos a informação do *id* da imagem e o limiar de IoU (linhas 12 – 13). Obtido o *bounding box* correto das predições, verificamos se o mesmo não é vazio e, então, caso não seja, armazenamos no conjunto de instâncias da imagem analisada. Repetimos este processo para todas as instâncias da imagem. (linhas 14–19). Por fim, salvamos o conjunto de instâncias encontradas na fusão na variável com informações da imagem corrente. Então, repetimos as mesmas análises para todas as imagens do conjunto de imagens classificadas (linhas 20–22). Finalmente, retornamos o conjunto com a fusão das predições (linha 23).

O Algoritmo 2 é uma função auxiliar usada no Algoritmo 1. Nele, recebemos como entrada, as coordenadas do *bounding box* que foi classificado, um conjunto de instâncias da mesma imagem predita pelos grupos específicos e um limiar de IoU. Inicialmente, declaramos uma variável para receber o *bounding box* final e uma variável para armazenar o maior valor para o calculo de IoU (linhas 2–3). Para cada instância dos *bounding boxes* preditos pelos grupos, calculamos o valor de IoU deste com o *bounding box* que foi classificado. Então, verificamos se este valor é maior ou igual ao nosso limiar de IoU, caso seja, atualizamo a

Algorithm 1 Algoritmo de fusão dos modelos específicos

Require: dados_classificados, predição_dos_grupos, limiar_de_confiança, limiar_de_iou.
Ensure: fusão_das_predições

```
1: function FUSÃO-DE-MODELOS
2:     fusão_das_predições ←{}
3:     for imagem em dados_classificados do
4:         instâncias_da_fusão ← []
5:         id_imagem_da_fusão ← id de imagem
6:         imagem_da_fusão ← {}
7:         imagem_da_fusão ← id_imagem_da_fusão
8:         for instância em imagem do
9:             if confiança da instância < limiar_de_confiança then
10:                continua
11:            end if
12:            classe_bbox ← classificação da instância
13:            bbox ← BUSCA-BBOX(bbox da instância, instâncias de predição_dos_grupos
    para a classe classe_bbox e com id id_imagem_da_fusão, limiar_de_iou)
14:            if bbox == [] then
15:                continue
16:            else
17:                instâncias_da_fusão ← bbox
18:            end if
19:        end for
20:        imagem_da_fusão ← instâncias_da_fusão
21:        fusão_das_predições ← imagem_da_fusão
22:    end for
23:    return fusão_das_predições
24: end function
```

variável que contém o *bounding box* de retorno e atualizamos o valor do nosso limiar de IoU (linhas 4–10). Por fim, retornos o valor do *bounding box* final encontrado.

5 Resultados Experimentais

Esta seção apresenta os resultados alcançados com os experimentos descritos na Seção 4. Ela contém duas partes de destaque: na primeira parte, reportamos os resultados obtidos nos experimentos para validar a hipótese de que modelos específicos funcionam melhor do que modelos gerais em um cenário de detecção de texto multilíngue. Na segunda parte, reportamos os resultados para o algoritmo, o qual tem o intuito realizar a fusão dos modelos específicos, visto que, independentemente das nossas conclusões para a primeira parte, ainda é necessário que a resposta final esteja agrupada em uma única solução.

Algorithm 2 Algoritmo de fusão dos modelos específicos

Require: bbox_classificado,bboxs_preditos, limiar_de_iou
Ensure: novo_bbox

```
1: function BUSCA-BBOX
2:     novo_bbox ← []
3:     max_iou ← limiar_de_iou
4:     for bbox_pred em bboxs_preditos do
5:         iou ← calcula iou entre bbox_classificado e bbox_pred
6:         if iou ≥ max_iou then
7:             novo_bbox ← bbox_pred
8:             max_iou ← iou
9:         end if
10:    end for
11:    return novo_bbox
12: end function
```

5.1 Modelo Geral × Modelo Específico

Esta subseção descreve o modelo geral e o modelo específico no cenário multilíngue explorado neste trabalho.

5.1.1 Experimento 1: Detecção de Texto Considerando um Treinamento Multilíngue.

A Tabela 1 mostra os resultados de desempenho considerando as imagens de teste para 10 idiomas. A partir dos resultados obtidos neste experimento e de uma análise dos casos de sucesso e falha deste modelo, pudemos observar que a PixelLink foi capaz de detectar várias regiões candidatas de texto parcialmente corretas.

Tabela 1: Resultados de desempenho da rede PixelLink treinados e avaliados com imagens contendo texto em 10 idiomas.

Precisão	**Revocação**	**Medida-F1**
67,73%	72,15%	69,87%

Em vários casos, os métodos realizaram uma detecção de um conjunto de palavras (Figura 9), o que diminuiu os resultados de precisão do método.

5.1.2 Experimento 2: Localização de Texto via Modelo Específico de Idioma.

A Tabela 2 mostra uma comparação entre os modelos estimados em cenários de treinamento multilíngue e específicos de idioma, considerando as imagens de teste contendo apenas texto no grupo especificado. Pode-se observar que o modelo específico de idioma supera o modelo

Figura 9: Exemplos de detecções preditas pelo método treinado em cenário multilíngue (verde) e sua respectiva detecção esperada (azul).

construído em um cenário de treinamento multilíngue para todos os grupos. Assim, podemos afirmar que, para resolver problemas de detecção de texto com vários idiomas, é possível obter melhores resultados utilizando um treinamento individual para detecção.

Tabela 2: Comparação dos resultados da rede PixelLink usando um modelo treinado com um grupo de idioma específico e um modelo treinado com todos os idiomas presente no conjunto de dados MLT. O conjunto de validação usado para obter as métricas da tabela corresponde as imagens do conjunto de validação MLT2019 com apenas as que contém o idioma do modelo específico usado, ou seja, Grupo 1 só foi comparado usando as imagem que contém latim.

	Modelo	**Precisão**	**Revocação**	**Medida-F1**
Grupo 1	Multilíngue	65,83	51,26	57,64
	Língua específica	74,78	54,12	62,80
Grupo 2	Multilíngue	46,67	46,20	46,43
	Língua específica	64,66	53,83	58,75
Grupo 3	Multilíngue	77,38	80,49	78,90
	Língua específica	84,95	84,12	84,53
Grupo 4	Multilíngue	77,32	66,19	71,33
	Língua específica	92,11	75,13	82,76

5.1.3 Avaliação Final da Proposta

Apesar da conclusão de que o uso de um modelo específico para as detecções de texto serem melhores do que um modelo geral (Tabela 2), quando consideramos o cenário de um

Tabela 3: Comparação dos resultados da rede PixelLink usando um modelo treinado com um grupo de idioma específico e um modelo treinado com todos os idiomas presentes no conjunto de dados MLT. O conjunto de validação usado para obter as métricas da tabela corresponde às imagens do conjunto de validação MLT2019.

Modelo	**Precisão**	**Revocação**	**Medida-F1**
Geral	**67.73**	**72.15**	**69.87**
Grupo 1	47,10	60,52	52,98
Grupo 2	65,74	63,48	64,59
Grupo 3	38,16	51,85	43,97
Grupo 4	45,12	59,69	51,39

conjunto de teste com todos os idiomas presentes na base MLT2019, os resultados não são tão satisfatórios. Como mostra a Tabela 3, o modelo geral possui a maior medida-F1 com 69,87%. Este resultado permite concluir que, no contexto de teste com todos os idiomas, não é possível utilizarmos um modelo específico, visto que o mesmo não poderá generalizar e acertar as detecções dos outros idiomas da qual ele não foi treinado. Portanto, para conseguirmos refletir o desempenho do modelo específico em um teste para o modelo geral, precisamos realizar uma fusão das detecções dos quatro modelos de idiomas específicos, e gerarmos um resultado integrado e compatível com o teste para um modelo geral.

5.2 Algoritmo para Fusão de Modelos Específicos de Idiomas

Nesta segunda parte da seção, reportaremos o resultado da solução utilizando o algoritmo mencionado na Subseção 4.2 da metodologia. Em nosso algoritmo, dois parâmetros que foram considerados são os limiares de confiança e IoU. Estes parâmetros são de suma importância, pois eles ditam quais *bounding boxes* serão utilizados ou não no conjunto de predições final. Desta forma, para entender como estes parâmetros se comportam em nosso algoritmo e para encontrar o melhor resultado da fusão, escolhemos uma faixa de valores para cada um destes limiares. Para a confiança, variamos seus valores entre 0,2, 0,4, 0,6, 0,8 e 0,95. Já o IoU, variamos entre 0,1, 0,3, 0,5 e 0,7. As Tabelas 4, 5 e 6 mostram os resultados obtidos nos experimentos com o algoritmo de fusão com todas as combinações dos limiares de entrada.

Para os valores obtidos na métrica de precisão, pode-se observar que, quanto maior os limiares, maior se torna o valor da precisão. Isto ocorre pois, com confianças maiores, selecionamos os *bounding boxes* com mais chances de estarem bem ajustados aos textos detectados. No caso do IoU, esta métrica garante que só consideremos *bounding boxes* com a maior correlação entre o *bounding box* encontrado no resultado do classificador. Por outro lado, quando observamos a métrica de revocação, pode-se ver um aumento quando diminuímos os valores dos limiares. Isso é possível pois os limiares menores permitem selecionarmos muitos *bounding boxes* para o conjunto final de fusão, o que pode acarretar uma imprecisão maior nos *bounding boxes* selecionados.

Dito isso, a melhor métrica para avaliarmos nossa melhor fusão é a medida-F1. Como

ela corresponde à média ponderada da precisão e da revocação, conseguimos selecionar a fusão que tem um melhor equilíbrio entre *bounding boxes* escolhidos e o ajuste do mesmo aos textos detectados. Portanto, nosso melhor valor para a medida-F1 se deu com uma confiança de 0.6 e um IoU de 0,5, obtendo assim, um valor de 81,67%.

Tabela 4: Resultado em termos de precisão para as avaliações do algoritmo de fusão de modelos específicos.

Precisão				
Confiança	**IoU**			
	0,1	0,3	0,5	0,7
0,2	80,15	85,06	93,09	96,78
0,4	85,08	88,76	94,80	97,71
0,6	88,11	91,05	95,94	98,34
0,8	90,52	92,87	96,88	98,73
0,95	93,12	94,80	97,90	**99,22**

Tabela 5: Resultado em termos de revocação para as avaliações do algoritmo de fusão de modelos específicos.

Revocação				
Confiança	**IoU**			
	0,1	0,3	0,5	0,7
0,2	**73.25**	73,09	72,18	62,55
0,4	72,74	72,62	71,72	62,30
0,6	72,06	71,93	71,10	61,95
0,8	70,51	70,43	69,72	61,13
0,95	66,52	66,46	65,94	58,86

Finalmente, na Tabela 7, encontramos a comparação final dos nossos experimentos. Com este resultado, conseguimos comprovar nossa hipótese inicial e concluirmos que, com um treinamento específico de idiomas, nosso modelo final conseguiu 11,8 pontos percentuais de melhoria em medida-F1 e 28,21 pontos percentuais a mais em precisão sobre um modelo que foi treinado com todos os idiomas.

6 Considerações Finais

Neste trabalho, no qual abordamos o tema de detecção de textos em cenário multilíngue, levantamos a hipótese inicial de que, ao realizarmos o treinamento de uma determinada arquitetura CNN e, no nosso caso de estudo, a PixelLink, em um cenário com apenas um

Tabela 6: Resultado em termos de medida-F1 para as avaliações do algoritmo de fusão de modelos específicos.

Medida-F1				
Confiança	**IoU**			
	0,1	0,3	0,5	0,7
0,2	76,54	78,62	81,32	75,98
0,4	78,43	79,88	81,66	76,09
0,6	79,28	80,37	**81,67**	76,01
0,8	79,28	80,11	81,09	75,51
0,95	77,60	78,14	78,80	73,89

Tabela 7: Comparativo final da avaliação do modelo geral contra o modelo específico com fusão das predições de cada grupo.

Modelo	**Precisão**	**Revocação**	**Medida-F1**
Língua específica	**95.94**	71.10	**81.67**
Multilíngue	67,73	**72,15**	69,87

idioma ao invés de todos, conseguiríamos obter melhores resultados para a precisão do nosso modelo, visto que nosso modelo estaria em contato apenas com dados de um idioma e, portanto, teria melhor desempenho para o mesmo.

Após um experimento inicial, pudemos comprovar nossa hipótese a partir de comparativos individuais das línguas, ou seja, comparamos um modelo que foi treinado para um cenário geral de detecção de texto multilíngue, com quatro modelos treinados considerando quatro grupos ou conjuntos de idiomas.

Para o Grupo 1, que compreende a língua latina apenas, o modelo específico conseguiu obter 8,95, 2,86 e 5,16 pontos percentuais a do mais que o modelo geral em precisão, revocação e medida-F1, respectivamente. Para o Grupo 2, que compreende a língua chinesa, japonesa e coreana, o modelo específico conseguiu obter 17,99, 7,63 e 12,32 pontos percentuais a do mais que o modelo geral em precisão, revocação e medida-F1, respectivamente. Para o Grupo 3, que compreende a língua bangla e hindi, o modelo específico conseguiu obter 7,57, 3,63 e 5,63 pontos percentuais a do mais que o modelo geral em precisão, revocação e medida-F1, respectivamente. Por fim, para o Grupo 4, que compreende a língua árabe apenas, o modelo específico conseguiu obter 14,79, 8,94 e 11,43 pontos percentuais a mais do que o modelo geral em precisão, revocação e medida-F1, respectivamente.

Estes resultados comprovaram nossa hipótese inicial, porém, em um cenário de teste real, ainda assim precisaremos fazer a detecção de texto multilíngue sem saber qual a língua que precisamos detectar. Portanto, desenvolvemos um algoritmo para realizar a fusão das detecções de cada grupo de idiomas, a fim de gerar um único resultado correspondente a nossa melhor combinação de *bounding boxes* vindos de cada um dos modelos específicos.

Com isso, obtivemos um resultado de 28,21 e 11,8 pontos percentuais de aumento em precisão e medida-F1, respectivamente. Nesta fusão final, nosso revocação ficou 1,05 pontos percentuais abaixo do modelo geral. Portanto, com este resultado, consolidamos a nossa hipótese como verdadeira e factível de ser implementada em um cenário real de teste para detecção multilíngue.

Os experimentos considerados neste trabalho são aplicáveis a qualquer arquitetura que aborde o problema de detecção de texto multilíngue, dado que expomos uma técnica de otimização de detecções neste cenário. Portanto, para trabalhos futuros, podemos aplicar a mesma técnica em outras arquiteturas para demonstrar se ela é invariante ao arcabouço usado. Além disso, nossa técnica considera um agrupamento de línguas baseado em semelhança morfológica, o que pode prejudicar a detecção de algum idioma em especifico que não é bem generalizado com um modelo construído para aquele grupo de línguas. Dessa forma, testar novas combinações de agrupamento de idiomas pode nos auxiliar a encontrar a melhor combinação de grupos, sem necessariamente usar todas as línguas separadamente. Além disso, é possível tratar as detecções mais recorrentes com técnicas e treinamento mais específicos, com o intuito de melhorar o resultado individual de um grupo especifico de línguas.

Finalmente, conseguimos aplicar os conhecimentos de aprendizado de máquina para investigar o problema de detecção de texto multilíngue, trazendo uma solução de otimização promissora para o cenário de visão computacional.

Referências

[1] George Awad, Asad Butt, Jonathan Fiscus, David Joy, Andrew Delgado, Martial Michel, Alan F. Smeaton, Yvette Graham, Wessel Kraaij, Georges Quénot, Maria Eskevich, Roeland Ordelman, Gareth J. F. Jones, e Benoit Huet. TRECVID 2017: Evaluating Ad-hoc and Instance Video Search, Events Detection, Video Captioning and Hyperlinking. Em *Proceedings of TRECVID 2017*. NIST, USA, 2017.

[2] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, F. Shafait, S. Uchida, e E. Valveny. Icdar 2015 competition on robust reading. Em *13th International Conference on Document Analysis and Recognition (ICDAR)*, páginas 1156–1160, Aug 2015.

[3] L. Li, R. Socher, e L. Fei-Fei. Towards total scene understanding: Classification, annotation and segmentation in an automatic framework. Em *IEEE Conference on Computer Vision and Pattern Recognition*, páginas 2036–2043, June 2009.

[4] Xiaoqing Liu e J. Samarabandu. An edge-based text region extraction algorithm for indoor mobile robot navigation. Em *IEEE International Conference Mechatronics and Automation, 2005*, volume 2, páginas 701–706 Vol. 2, July 2005.

[5] Sergi Garcia-Bordils, George Tom, Sangeeth Reddy, Minesh Mathew, Marçal Rusiñol, C. V. Jawahar, e Dimosthenis Karatzas. Read while you drive - multilingual text tracking on the road. Em Seiichi Uchida, Elisa Barney, e Véronique Eglin, editors,

Document Analysis Systems, páginas 756–770, Cham, 2022. Springer International Publishing.

[6] A. Mammeri, A. Boukerche, e E. H. Khiari. Mser-based text detection and communication algorithm for autonomous vehicles. Em *IEEE Symposium on Computers and Communication (ISCC)*, páginas 1218–1223, June 2016.

[7] Q. Ye e D. Doermann. Text detection and recognition in imagery: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(7):1480–1500, July 2015.

[8] Shangbang Long, Siyang Qin, Dmitry Panteleev, Alessandro Bissacco, Yasuhisa Fujii, e Michalis Raptis. Towards end-to-end unified scene text detection and layout analysis. Em *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, páginas 1049–1059, June 2022.

[9] Dan Deng, Haifeng Liu, Xuelong Li, e Deng Cai. PixelLink: Detecting Scene Text via Instance Segmentation. *ArXiv*, abs/1801.01315, 2018.

[10] Sheng Zhang, Yuliang Liu, Lianwen Jin, e Canjie Luo. Feature enhancement network: A refined scene text detector, 2017. URL `https://arxiv.org/abs/1711.04249`.

[11] M. Liao, B. Shi, e X. Bai. Textboxes++: A single-shot oriented scene text detector. *IEEE Transactions on Image Processing*, 27(8):3676–3690, Aug 2018.

[12] Jianqi Ma, Weiyuan Shao, Hao Ye, Li Wang, Hong Wang, Yingbin Zheng, e Xiangyang Xue. Arbitrary-oriented scene text detection via rotation proposals. *IEEE Transactions on Multimedia*, 20(11):3111–3122, nov 2018.

[13] Xiang Li, Wenhai Wang, Wenbo Hou, Ruo-Ze Liu, Tong Lu, e Jian Yang. Shape robust text detection with progressive scale expansion network, 2018. URL `https://arxiv.org/abs/1806.02559`.

[14] Kiran Perveen, Rukhsana Perveen, e Dr. Awais Yasin. Survey of multilingual script identification techniques on wild images. *LC International Journal of STEM*, 3(1): 1–14, Apr. 2022.

[15] Max Jaderberg, Andrea Vedaldi, e Andrew Zisserman. Deep features for text spotting. Em David Fleet, Tomas Pajdla, Bernt Schiele, e Tinne Tuytelaars, editors, *European Conference on Computer Vision (ECCV)*, páginas 512–528, Cham, 2014. Springer International Publishing.

[16] Weilin Huang, Yu Qiao, e Xiaoou Tang. Robust scene text detection with convolution neural network induced mser trees. Em *ECCV*, 2014.

[17] Shaoqing Ren, Kaiming He, Ross Girshick, e Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 28, 2015.

[18] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, e Alexander C. Berg. SSD: Single shot MultiBox detector. Em *European Conference on Computer Vision (ECCV)*, páginas 21–37. Springer International Publishing, 2016.

[19] Joseph Redmon, Santosh Divvala, Ross Girshick, e Ali Farhadi. You only look once: Unified, real-time object detection. Em *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, páginas 779–788, 2016.

[20] Alexander Neubeck e Luc Van Gool. Efficient non-maximum suppression. Em *18th International Conference on Pattern Recognition - Volume 3*, ICPR '06, páginas 850–855, Washington, DC, USA, 2006. IEEE Computer Society.

[21] Minghui Liao, Baoguang Shi, Xiang Bai, Xinggang Wang, e Wenyu Liu. Textboxes: A fast text detector with a single deep neural network, 2016. URL `https://arxiv.org/abs/1611.06779`.

[22] Xiqi Wang, Shunyi Zheng, Ce Zhang, Rui Li, e Li Gui. R-yolo: A real-time text detector for natural scenes with arbitrary rotation. *Sensors*, 21(3), 2021.

[23] Yuliang Liu, Hao Chen, Chunhua Shen, Tong He, Lianwen Jin, e Liangwei Wang. Abcnet: Real-time scene text spotting with adaptive bezier-curve network. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, páginas 9806–9815, 2020.

[24] Yuxin Wang, Hongtao Xie, Zheng-Jun Zha, Mengting Xing, Zilong Fu, e Yongdong Zhang. Contournet: Taking a further step toward accurate arbitrary-shaped scene text detection. Em *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[25] Jianqi Ma, Weiyuan Shao, Hao Ye, Li Wang, Hong Wang, Yingbin Zheng, e Xiangyang Xue. Arbitrary-oriented scene text detection via rotation proposals. *IEEE Transactions on Multimedia*, 20(11):3111–3122, 2018.

[26] Pengyuan Lyu, Minghui Liao, Cong Yao, Wenhao Wu, e Xiang Bai. Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43:532–548, 2021.

[27] Xiang Li, Wenhai Wang, Wenbo Hou, Ruo-Ze Liu, Tong Lu, e Jian Yang. Shape robust text detection with progressive scale expansion network. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, páginas 9328–9337, 2019.

[28] Wenhai Wang, Enze Xie, Xiaoge Song, Yuhang Zang, Wenjia Wang, Tong Lu, Gang Yu, e Chunhua Shen. Efficient and accurate arbitrary-shaped text detection with pixel aggregation network. *IEEE/CVF International Conference on Computer Vision (ICCV)*, páginas 8439–8448, 2019.

[29] Minghui Liao, Yan Wan, Cong Yao, Kai Chen, e Xiang Bai. Real-time scene text detection with differentiable binarization. *AAAI Conference on Artificial Intelligence*, 34:11474–11481, 04 2020.

[30] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, F. Shafait, S. Uchida, e E. Valveny. ICDAR 2015 Competition on Robust Reading. Em *13th International Conference on Document Analysis and Recognition*, páginas 1156–1160, August 2015.

[31] Nibal Nayef, Fei Yin, Imen Bizid, Hyunsoo Choi, Yuan Feng, Dimosthenis Karatzas, Zhenbo Luo, Umapada Pal, Christophe Rigaud, Joseph Chazalon, Wafa Khlif, Muzzamil Luqman, Jean-Christophe Burie, Cheng-Lin Liu, e Jean-Marc Ogier. ICDAR 2017 Robust Reading Challenge on Multi-Lingual Scene Text Detection and Script Identification - RRC-MLT. Em *14th IAPR International Conference on Document Analysis and Recognition*, páginas 1454–1459, 11 2017.

[32] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, e Hwalsuk Lee. Character Region Awareness for Text Detection. Em *IEEE Conference on Computer Vision and Pattern Recognition*, páginas 9365–9374, 2019.

[33] Andreas Veit, Tomas Matera, Lukas Neumann, Jiri Matas, e Serge Belongie. Coco-text: Dataset and benchmark for text detection and recognition in natural images. Em *arXiv preprint arXiv:1601.07140*, 2016.

[34] Nibal Nayef, Yash Patel, Michal Busta, Pinaki Nath Chowdhury, Dimosthenis Karatzas, Wafa Khlif, Jiri Matas, U. Pal, Jean-Christophe Burie, Cheng lin Liu, e Jean-Marc Ogier. ICDAR2019 Robust Reading Challenge on Multi-lingual Scene Text Detection and Recognition - RRC-MLT-2019. *ArXiv*, abs/1907.00945, 2019.

[35] Ross Girshick, Jeff Donahue, Trevor Darrell, e Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. Em *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[36] Ross Girshick. Fast r-cnn. Em *IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[37] Jhonatas Conceição, Allan Pinto, Luis Decker, Jose Luis Campana, Manuel Neira, Andrezza dos Santos, Helio Pedrini, e Ricardo Torres. Multi-lingual text localization via language-specific convolutional neural networks. Em *Anais Estendidos da XXXII Conference on Graphics, Patterns and Images*, páginas 215–218, Porto Alegre, RS, Brasil, 2019. SBC.

[38] A. Shrivastava, A. Gupta, e R. Girshick. Training Region-Based Object Detectors with Online Hard Example Mining. Em *IEEE Conference on Computer Vision and Pattern Recognition*, páginas 761–769, June 2016.