Optical Character Recognition usando Deep Learning

Claudio Filipi Gonçalves dos Santos,
Fabricio Aparecido Breve
Instituto de Geociências e Ciências Exatas - IGCE - UNESP
Rio Claro, Brasil
csantos@rc.unesp.br, fabricio@rc.unesp.br

***Resumo*—O objetivo desse projeto é adaptar e combinar técnicas de Deep Learning para realizar várias etapas de um software de OCR, com a capacidade de detectar áreas com caracteres e transcrever essas regiões em texto. A hipótese é de que tais técnicas possam melhorar a acurácia da tarefa de OCR em textos livres, da mesma forma que ocorreu em outras tarefas onde foi aplicada.**

Inteligência Computacional

I. Introdução

Optical Character Recognition (OCR) é o nome dado à tecnologia utilizada para extrair e identificar caracteres de imagens, de modo que possam ser salvos como um arquivo de texto. Este problema já foi bastante abordado nas últimas décadas, o OCR feito a partir de textos bem definidos, como em livros, jornais e revistas, normalmente apresenta uma acurácia alta. Porém, o OCR feito para textos livres, a partir de placas de trânsito, anúncios em cartazes, onde há uma grande variação no tamanho, cor e outros aspectos das letras, ainda é uma tarefa bastante desafiadora.

Deep learning é um ramo da área de aprendizado de máquinas baseado em um conjunto de algoritmos que visa modelar em alto nível as abstrações de dados usando várias camadas de processamento com complexas estruturas, compostas de transformações não-lineares.

II. Conceitos e Técnicas

O primeiro passo do framework, que parece ser o mais dificil de todos, é a detecção de áreas candidatas à texto. Algumas técnicas já mostraram alguma eficácia na execução dessa tarefa enquanto outras, apesar de já terem demonstrado grande eficácia em trabalhos parecidos com o OCR, ainda não foram testadas no contexto de detecção e segmentação de caracteres em imagens, sejam elas do mundo real ou de documentos que contem apenas texto.

Com a intenção de desenvolver um *framework* usando apenas técnicas de *Deep Learning*, será desenvolvido uma adaptação em uma técnica já conhecida chamada *YOLO - You Only Look Once* [1]. À primeira vista essa técnica parece não ser muito complexa de ser implementada, pois essa rede neural consiste apenas de camadas convolucionais seguidas de *pooling* 2x2 até uma última camada que ao invés de ser um *perceptron* de múltiplas camadas para classificação, é formada por um cubo convolucional que determina as áreas de interesse em cima de uma imagem.

Os problemas começam quando é necessário um treinamento específico, como será o caso desse trabalho. Os problemas podem ser bem divididos nas seguintes partes:

- Cálculo da perda: redes neurais do tipo *Deep Learning* costumam fazer o cálculo de perda (mais conhecida como *loss*) usando Erro Médio Quadrático (*Mean Square Error - MSE*) para sistemas com o intuito de regressão ou Entropia Cruzada de Categorias (*Categorical Cross Entropy*) quando é desejado uma classificação. Porem nesse trabalho será desejado que a rede faça a detecção de áreas de interesse, normalmente modelada como um regressor. Nesse caso, para aumentar a precisão da área de interesse, será necessário usar a métrica Intersecção sobre União (*Intersection over Union - IoU*) que como o nome diz, cálcula a razão entre as áreas de intersecção predita pela verdadeira dividida pela área de união predita com a verdadeira;
- Qual área é relativa a que parte da imagem: é possível ver no trabalho relacionado ao *YOLO* [1] que em uma mesma imagem são detectadas dezenas ou centenas de áreas de interesse e são mostradas apenas as áreas com pontuação de confiança acima de um certo valor (normalmente 0.3). No período de treinamento, mesmo sabendo em que áreas estão as partes de interesse da imagem, será necessário reorganizar esses dados usando Supressão Não-Máxima (*Non-Maximun Supression - NMS* [2]), um algoritmo que reordena essas áreas de acordo com o IoU.

Após a detecção de áreas candidatas, o *framework* proposto neste projeto visa usar redes neurais do tipo Deep Learning usando uma combinação de *Convolutional Neural Network (CNN)* com redes neurais recorrentes (*Recurrent Neural Network - RNN*) gerando uma rede híbrida que visa transformar a área detectada em uma série temporal. Esse tipo de combinação já foi usada anteriormente [3] usando certos

VII Workshop do Programa de Pós-Graduação em Ciência da Computação: "Interação entre Academia e Empresa", Unesp, Rio Claro, 14 e 15 de setembro de 2017.

"Bolsista CAPES"

A placa de vídeo Quaddro M5000 usada nessa pesquisa foi doada pela NVIDIA Corporation.

tipos de redes recorrentes. a intenção desse trabalho é usar outros tipos ainda não testados.

III. Metodologia de Desenvolvimento

O framework proposto pelo candidato possui duas maiores dificuldades: a detecção de áreas candidatas e a transformação dessa área em texto. A Figura 1 resume em alto nível como deverá funcionar o sistema.

Pensando no primeiro problema, há fortes indícios de que o uso de uma rede neural parecida com a *YOLO* [1] seja capaz de detectar os pontos da imagem que contenha texto.

Partindo para a segunda parte do problema, já são conhecidas algumas soluções usando a combinação de CNN com redes recorrentes do tipo *Long Shor Term Memory - LSTM* [3]. Visando inovar essa estrutura, haverá a substituição da camada LSTM por uma camada *Gated Recurrent Unit - GRU*, um outro tipo de rede recorrente com algumas diferenças estruturais.

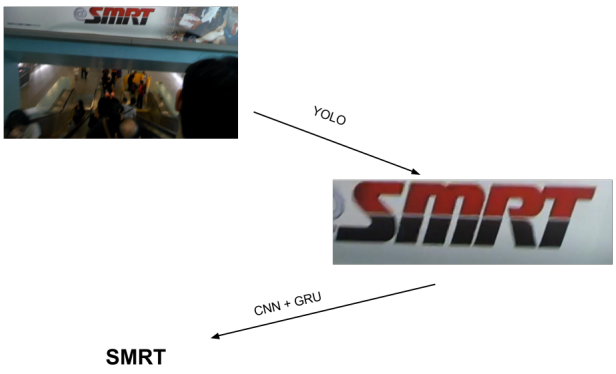

Figura 1. Expectativa de como funcionará o trabalho final

Para testar os algoritmos a serem desenvolvidos e compará-los com outras técnicas, será usada a base de dados do ICDAR - Challenge 1 - Text Localization. Como comparativo, serão usadas as seguintes métricas, seguidas de uma breve descrição de cada:

- Precision: acertos/(acertos + falsos positivos);
- Recall: acertos/(acertos + falsos negativos);
- Harmonic Mean: cálculo da diferença entre a área detectada e a área que realmente delimita a região [25].

IV. Resultados Preliminares

Esse trabalho previa anteriormente a divisão da tarefa principal em três partes: detecção de áreas com texto nas imagens, segmentação dessa área em caracteres e a classificação desse segmentos em caracteres em definitivo. Visando uma alta taxa de acerto na classificação, foi feita a inscrição em uma competição de aprendizagem de máquina que visava justamente a classificação de caracteres de imagens do mundo real. O resultado obtido na competição foi considerada um sucesso, já que houve um índice de acertos muito alto. A seguinte arquitetura foi usada:

- Entradas de tamanho 32x32, em preto-e-branco;
- Dupla camada convolucional 3x3 com 128 camadas, seguida de Pooling 2x2;
- Dupla camada convolucional 3x3 com 256 camadas, seguida de Pooling 2x2;
- Dupla camada convolucional 3x3 com 512 camadas, seguida de Pooling 2x2;
- Camada Multilayer Perceptron com 4096 neurônios, com Dropout de 75 por cento e Rectfied Linear Unit como saida;
- Camada Multilayer Perceptron com 4096 neurônios, com Dropout de 75 por cento e Rectfied Linear Unit como saida;
- Classificador *Logistic Regression* com 62 classes(A-Z, a-z, 0-9)

Usando a competição “First Steps with Julia“ do site Kaggle [4] como plataforma de avaliação des resultados iniciais, foi alcançado 39,4 por cento de acertos com uma outra arquitetura mais simples. A combinação da arquitetura descrita acima com a técnica de aumento artificial dos dados alcançou um acerto de 83,5 por cento, colocando esse trabalho na quarta posição.

V. Considerações Finais

O trabalho encontra-se atualmente em fase de implementação. Foi feito um profundo estudo sobre as técnicas e neste momento as primeiras implementações estão sendo feitas usando a linguagem Python com o framework Keras [5].

Referências

[1] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” *CoRR*, vol. abs/1506.02640, 2015. [Online]. Available: http://arxiv.org/abs/1506.02640

[2] J. H. Hosang, R. Benenson, and B. Schiele, “Learning non-maximum suppression,” *CoRR*, vol. abs/1705.02950, 2017. [Online]. Available: http://arxiv.org/abs/1705.02950

[3] A. Karpathy and F. Li, “Deep visual-semantic alignments for generating image descriptions,” *CoRR*, vol. abs/1412.2306, 2014. [Online]. Available: http://arxiv.org/abs/1412.2306

[4] “First steps with julia,” https://www.kaggle.com/c/street-view-getting-started-with-julia, 2014.

[5] F. Chollet *et al.*, “Keras,” https://github.com/fchollet/keras, 2015.