TP2 - Reconhecimento de CAPTCHA

Izabela Tavares e Beatriz Reis

Novembro 2023

1 Introdução

Este trabalho consiste na descrição e análise dos resultados da implementação de dois métodos distintos para o reconhecimento de CAPTCHA. Enquanto o primeiro associa o descritor de características HOG à classificação com algoritmos de aprendizado de máquina, o segundo método consiste em uma rede neural convolucional. Neste relatório, são destrinchadas as decisões relativas a ambos os métodos e apresentados os resultados obtidos em cada um deles, além de comparadas as implementações.

2 Histogram of Gradients + Classificador

Para o uso de HOG associado a um classificador, foram elaboradas algumas arquiteturas diferentes. São elas: Regressão Logística, SVM (Support Vector Machine) com otimização de hiperparâmetros por meio de Grid Search, KNN (K-Nearest Neighbors) e Ensemble por meio de Stacking utilizando os modelos XGBoost, Random forest e Regressão Logística. Apesar de apresentarem algumas diferenças entre si, essas implementações têm em comum o pipeline do método HOG + Classificador, que pode ser descrita como:

1. Pré-processamento: remoção de ruído em toda a imagem, visando à maior precisão das características extraídas pelo HOG;
2. Segmentação: divisão da imagem de entrada em 6 imagens, cada uma contendo um caractere;
3. Extração das características de cada imagem (HOG):
   (a) Cálculo do gradiente;
   (b) Cálculo do histograma de gradientes;
   (c) Normalização dos valores;
4. Classificação: a partir de vetores contendo as características extraídas pelo HOG e uma label associada (one-hot encoded), foi feito o treinamento do modelo escolhido, sua validação e, por fim, seu teste.

Para a KNN, foram feitos testes com valores de K (quantidade de vizinhos) entre 0 e 200 e, nos resultados (Tabela 1), são apresentados os 3 melhores resultados obtidos, com K entre 2 e 4. A Figura 1 apresenta a relação entre a escolha de K e a acurácia do método KNN.

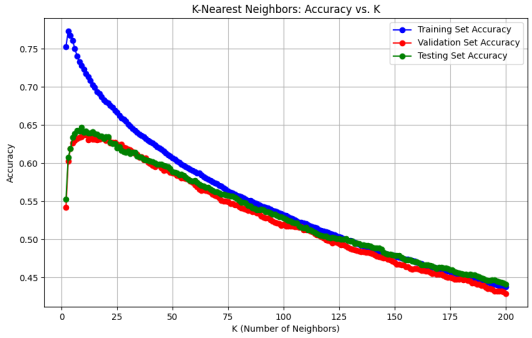


Figura 1: Relação entre escolha de K e acurácia do método KNN.

	Acc (treinamento)	**Acc (validação)**	**Acc (teste)**	**Tempo de execução**
Regressão Logística	1.0	0.85	0.85	30 min
Ensemble	0.75	0.75	0.79	+11h
SVM	0.80	0.74	0.72	5h
KNN (K=4)	0.77	0.62	0.62	25 min
KNN (K=3)	0.77	0.60	0.61	25 min
KNN (K=2)	0.75	0.54	0.55	25 min

Tabela 1: Resultados de cada método, ordenados por acurácia no teste.

A partir dos resultados apresentados na Tabela 1, considerando tanto o custo quanto a acurácia, percebe-se a significativa superioridade da classificação por Regressão Logística, que apresentou uma alta eficiência em baixo tempo de execução. Enquanto isso, os métodos de Ensemble e SVM resultaram em alta acurácia, porém têm execução bastante lenta. O método KNN, por sua vez, é executado em tempo similar à Regressão e gera bons resultados, mas ainda abaixo da Regressão. Na Figura 3a, percebe-se, conforme esperado, que a taxa de reconhecimento decai em função do número mínimo de caracteres reconhecidos, e, além disso, que a taxa de acerto do método Regressão Logística é alta, conforme revelado pela medida de acurácia do modelo.

3 Convolutional Neural Network

A implementação de redes neurais convolucionais para o reconhecimento dos caracteres do CAPTCHA pode ser descrita pelo pipeline:

1. Pré-processamento: remoção de ruído em toda a imagem, visando à maior precisão das características extraídas pelo HOG;
2. Segmentação: divisão da imagem de entrada em 6 imagens, cada uma contendo um caractere;
3. Definição da arquitetura da CNN;
4. Treinamento do modelo;
5. Avaliação do modelo.

A arquitetura da CNN consiste, em geral, em três camadas convolucionais, utilizando a função ReLU para ativação, com Max Pooling entre elas, além de duas camadas densas: a primeira utilizando a função ReLu e a segunda a Sigmoid. Para além dessa estrutura, foi testada também a utilização de dropout de 50% entre as camadas densas e a variação na quantidade de épocas: foram testadas uma rede com 15 épocas e dropout e outra com 60 épocas com e sem dropout. A Tabela 2 apresenta os resultados dessas diferentes arquiteturas de CNN.

Acurácia/ Perda	**Treinamento**	**Validação**	**Teste**
15 épocas (com dropout)	0.97/0.08	0.94/0.19	0.94/0.20
60 épocas (com dropout)	0.99/0.01	0.94/0.25	0.94/0.29
60 épocas (sem dropout)	0.99/0.01	0.88/0.68	0.82/0.75

Tabela 2: Resultados da variação na CNN.

É importante destacar que iniciamos com 60 épocas e sem dropout para reduzir a diferença entre acurácia nos treinos e validação/teste. Após adicionar dropout, houve melhorias, mas os gráficos mostraram oscilação após 15 épocas, indicando possível acúmulo de erro (ver Figura 2). Por esse motivo, optamos por reduzir a quantidade de épocas, o que se mostrou mais eficiente. Na Figura 3b, percebe-se ser bastante alta a taxa de acerto da arquitetura com 15 épocas e dropout, conforme era esperado, dada a alta acurácia dos resultados (94%).

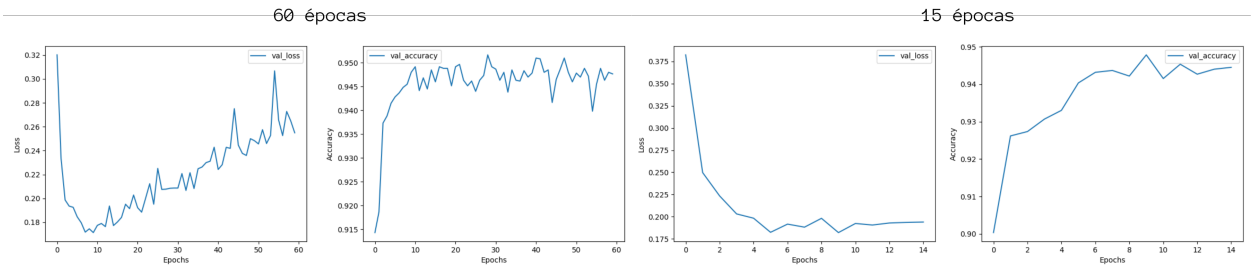
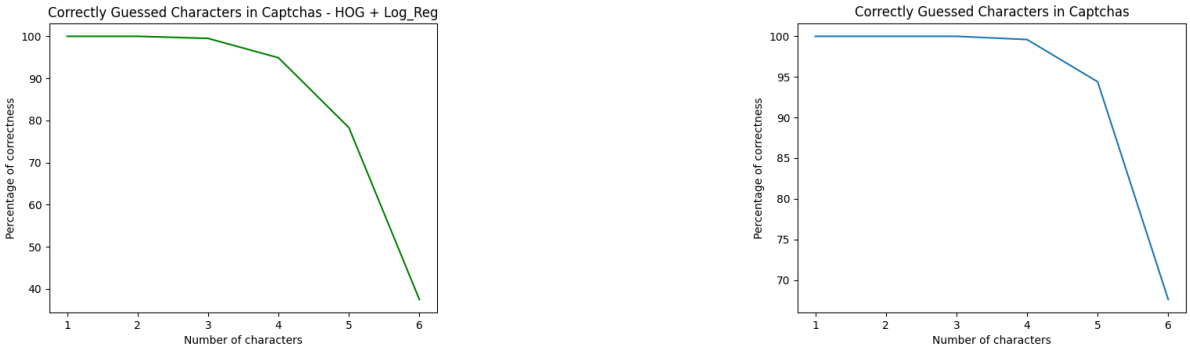


Figura 2: Perda e acurácia no conjunto de validação 60 épocas vs. 15 épocas.



(a) HOG + Regressão Logística

(b) CNN

Figura 3: Comparação da taxa de reconhecimento em ambos os métodos apresentados.

4 Conclusão

Conforme indicado pelas taxas de reconhecimento apresentadas na Figura 3 e pela acurácia de cada método, tanto o uso de HOG + Classificador quanto a CNN apresentam bons resultados. No entanto, em geral, a utilização de redes convolucionais é mais eficiente, tanto em termos de custo quanto de resultados.

Abaixo, na Figura 4, estão alguns resultados para as entradas de teste, destacando casos em que 6, 5, 4 e 3 letras foram previstas corretamente.

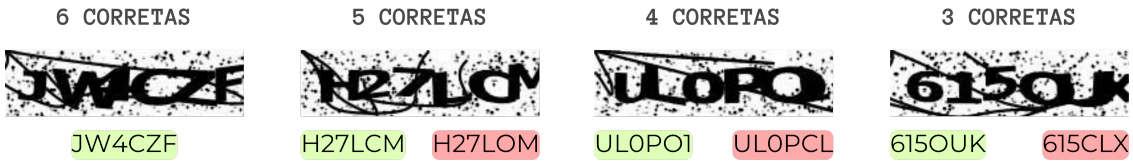


Figura 4: Resultados reais em casos de teste.

Referências

[1] CAPTCHA Recognition using Convolutional Neural Network. Disponível em: `https://medium.com/@manvi./captcha-recognition-using-convolutional-neural-network-d191ef91330e`

[2] HOG (Histogram of Oriented Gradients): An Overview. Disponível em: `https://towardsdatascience.com/hog-histogram-of-oriented-gradients-67ecd887675f`