

Reconhecimento do ID das Placas Destinadas ao Enfornamento do Laminador de Tiras a Quente Utilizando Deep Learning

Erick Ramos dos Santos¹, Gustavo Maia de Almeida²

^{1,2}Instituto Federal do Espírito Santo – Campus Serra, Espírito Santo, Brasil

rdserick@gmail.com¹, gmaia@ifes.edu.br²

Resumo. Este artigo apresenta um método de reconhecimento, por meio de visão computacional, do ID das placas direcionadas aos fornos do LTQ (laminador de tiras a quente) de uma Siderurgia, sendo coletadas imagens destas placas através de uma câmera posicionada antes da entrada dos fornos. O atual sistema da empresa não consegue se manter eficaz durante todo o período do dia devido à dificuldade de extrair características das imagens com iluminação não-uniforme. Com isso, um novo método é proposto, realizando um pré-tratamento com filtro de mediana e filtro gaussiano e, posteriormente, realizando binarização. As imagens tratadas são direcionadas a uma rede neural *Deep Learning* com arquitetura convolucional visando extrair as características e classificar os dígitos. Todo o desenvolvimento do projeto foi feito através do *software* Matlab, sendo o banco de dados de treinamento da rede neural construído com dígitos cortados das imagens das placas. O novo código obteve resultados significativos, com uma taxa de acerto de 99,78% ao utilizar um banco de dados com mil e oitenta imagens de cada dígito, totalizando dez mil e oitocentas imagens para o treinamento.

Palavras-Chave: Visão Computacional, *Deep Learning*, Convolucional.

1 Introdução

O processo de laminação de tiras a quente (LTQ) de uma Siderurgia produtora de aço consiste na etapa de acabamento da laminação de chapas finas produzindo bobinas. A placa de aço proveniente do forno de reaquecimento é direcionada à trens acabadores posicionados uns em frente aos outros, realizando deformações de forma contínua, sendo esta sequência realizada em resfriamento contínuo. A espessura das placas é reduzida e conseqüentemente o comprimento é aumentado. Ao término do processo, uma placa que possuía espessura entre 225 mm e 250 mm e comprimento entre 4 m e 11 m, transforma-se em uma bobina com espessura entre 1 mm e 20 mm e comprimento entre 80 m e 1 km, sendo ambas as propriedades definidas pelo cliente. A figura 1 ilustra uma representação esquemática do processo.

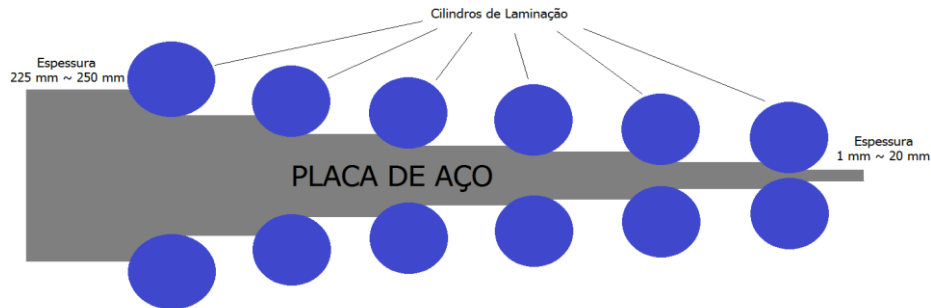


Fig. 1. Representação esquemática do processo do LTQ

O controle do pátio de placas do LTQ é feito através de um sistema semi-automatizado que, ao receber as informações específicas de cada placa, consegue movê-las de maneira inteligente e eficaz. O tipo de material, a espessura e o comprimento da placa, são exemplos destas informações. O recebimento destes dados faz-se necessário para realizar a movimentação das placas, pois, se baseando no pedido da bobina do cliente, o software decide qual placa deve ser enfiada, enviando assim, o comando ao operador.

Ao serem formadas no lingotamento contínuo, as placas recebem a escrita do seu respectivo ID em sua superfície, como representado na figura 2.



Fig. 2. ID escrito na placa

Ao sair do lingotamento contínuo, existem três destinos possíveis para a placa: pátio de placas do LTQ, condicionamento (em caso de placas com defeito que precisam ser recondicionadas) e o pátio de placas destinadas para venda.

Quando as placas são direcionadas à laminação de tiras a quente para a formação das bobinas, uma câmera posicionada antes da entrada dos fornos fotografa a escrita da placa que está sendo enfiada, visando confirmar se condiz o ID informado pelo sistema. Essa averiguação é uma forma de controlar o processo e verificar sua confiabilidade, visto que se uma placa for enfiada de forma equivocada, a bobina produzida terá propriedades diferentes das exigidas pelo cliente, causando um atraso na produção da empresa e, conseqüentemente, um prejuízo notável.

Tendo em vista a forma de operação do sistema, o ID da placa é uma informação única e vital para o bom funcionamento de todo o processo da indústria. Logo, buscando uma maneira automática de realizar a confirmação do ID, um software de reconhecimento foi criado por meio de visão computacional. Desta forma, elimina a necessidade da presença de um operador durante todo o dia, visto que o processo trabalha de maneira 24/7 (24 horas por dia, 7 dias por semana), desconsiderando o tempo perdido devido a paradas obrigatórias.

Este artigo está dividido em cinco seções. A seção 2 descreve em mais detalhes a problematização do processo de leitura dos ID's das placas. A seção 3 apresenta a metodologia adotada e os seus desafios. Os resultados obtidos são apresentados na seção 4, e por fim, a seção 5 apresenta as conclusões do estudo e propõe um caminho futuro de pesquisa.

2 Problematização

Atualmente, o sistema de reconhecimento não consegue se manter eficaz durante todo o período do dia, devido à dificuldade no controle da iluminação do amplo local onde estão situados os fornos. Outros problemas, como a falta de padrão de escrita nos lingotamentos contínuos e o vapor proveniente de uma placa com alta temperatura, também são válidos de serem citados, porém o último caso é desconsiderado, tendo em vista que identificar o ID nesta situação se torna algo impossível ao olho nu. O atual sistema trabalha com pré-processamento da imagem tendo um algoritmo que a binariza através de *Adaptive Threshold* [1] [2], buscando sempre manter as bordas dos dígitos, visto que os números serão extraídos da imagem utilizando *Tesseract*, um *software* de OCR (Reconhecimento Óptico de Caracteres) de código aberto [3] que imprime os números de uma imagem em um arquivo txt. Ao ocorrer os problemas citados, realizar um pré-processamento via *software* em que a definição das bordas não é perdida, torna-se algo inviável a ser trabalhado, como ilustrado na figura 3, tornando o classificador ineficaz, possuindo uma taxa de acertos durante o dia em torno de 88%.



Fig. 3. Exemplo de imagem recebida com iluminação ruidosa seguido da binarização feita através de *Adaptive Threshold*

Portanto, torna-se necessário uma nova abordagem para a realização do reconhecimento do ID, de tal maneira que os obstáculos sejam vencidos via codificação.

Buscando vencer a não uniformidade do ambiente, o presente trabalho propõe um pré-tratamento das imagens, utilizando filtro de mediana e filtro gaussiano para diminuição de ruídos na binarização das mesmas. Juntamente com a melhoria no tratamento da imagem, é proposto um diferente classificador construído com *Deep Learning* tendo como arquitetura uma rede convolucional para extrair as características dos dígitos [4]. Desta forma, espera-se que o novo *software* seja mais versátil aos problemas, se mantendo eficaz durante todo o período do dia.

3 Metodologia

Com a finalidade de obter uma melhor classificação do ID de cada placa, torna-se necessário uma binarização eficiente para que o sistema projetado possa reconhecer cada dígito e enviá-los ao classificador. Ao binarizar uma imagem com qualidade comprometida, a probabilidade de ruídos no resultado é alta, afetando diretamente o reconhecimento dos dígitos. Visando a atenuação destes ruídos, foram projetados filtros que auxiliam neste tratamento.

Como pode ser visto na figura 1, cada ID é composto por uma letra e nove dígitos. A letra representa a origem da placa, ou seja, qual dos três lingotamentos contínuos existentes na siderurgia produziu a placa, sendo esta informação irrelevante nesta parte do processo. Com isso, ao reconhecer a letra, o código irá ignorá-la, aumentando a eficiência do classificador, pois desta forma, evita a possibilidade de o classificador ‘confundir’ letras e dígitos semelhantes, como por exemplo, ‘B’ e ‘8’.

Todo o projeto foi codificado com o auxílio do *software* Matlab que possui ferramentas para projetar todos os métodos propostos.

3.1 Codificação

A binarização é um método de segmentação de imagens que consiste em converter uma imagem com níveis de cinza para uma representação binária. Esta técnica possibilita identificar objetos e separá-los do fundo da imagem, característica importante ao projeto, visto que deseja-se separar cada dígito para que seja extraídas suas características e, conseqüentemente, classificado.

As imagens captadas pela câmera estão em RGB, ou seja, possuem três dimensões de cores aditivas, sendo elas: vermelho, verde e azul. Logo, o primeiro passo é a conversão da imagem em RGB para escala de cinza. Após a conversão, o próximo passo consiste na aplicação dos filtros para redução dos ruídos, tendo em vista que a aplicação da binarização sem um tratamento correto da imagem tende a ser algo ineficaz.

O filtro de mediana permite a atenuação dos ruídos da imagem através da ordenação dos níveis de cinza em ordem crescente de intensidade [5]. O ponto negativo deste filtro é a eliminação de linhas consideravelmente finas, o que não se torna algo prejudicial ao presente trabalho.

Buscando a suavização da imagem, a aplicação do filtro gaussiano torna-se uma alternativa, visto que se trata de um filtro passa-baixa, eliminando os valores de alta frequência como em [6]. Sendo uma suavização mais delicada, o filtro de média, consegue preservar mais os contornos da imagem, conseguindo assim, uma melhora na imagem sem perdas de informações importantes.



Fig. 4. Binarização da imagem com aplicação dos filtros

Após realizado o pré-tratamento na imagem, e posteriormente sua binarização, como ilustrado na figura 4, torna-se possível o reconhecimento dos dígitos. Para separar cada dígito é utilizada a função do Matlab chamada 'bwlable', que retorna todos os componentes reconhecidos na imagem binarizada. Buscando o corte de cada dígito para que sejam submetidos ao classificador, utiliza-se a função 'regionprops', na qual obtém informações de cada componente. Recebendo a informação 'BoundingBox' desta função, as seguintes características dos componentes são obtidas: as coordenadas do ponto inicial, largura e altura. Com essas informações, utilizando a função 'imcrop', a letra e os nove dígitos são cortados e separados em uma célula.

Como já citado, a letra é irrelevante nesta parte do processo, logo, será eliminada, restando apenas os dígitos a serem reconhecidos. É preciso ter cuidado ao realizar o corte dos números, visto que alguns dígitos estão muito próximos, podendo ser reconhecidos dois dígitos como apenas um, como pode ser visto nos dois últimos números da figura 3. Visando solucionar este problema, foi criada uma lógica condicional com a área dos dígitos, buscando reconhecer quando acontece este equívoco, podendo assim, partir em dois dígitos se baseando no centroide do componente, como mostra a figura 5.



Fig. 5. Números recortados da imagem binarizada

3.2 Classificador *Deep Learning*

Todo sistema que envolve reconhecimento de padrões necessita de um classificador bem configurado, pois este que irá rotular cada informação de entrada. A rede neural

artificial do tipo *Deep Learning* é uma técnica eficaz, com grande precisão, que atualmente, é considerado o estado da arte na área de visão computacional, possuindo ótimos resultados na classificação de imagens [7].

O *Deep Learning* é um método de *Machine Learning* que possui a capacidade de aprendizagem em grandes quantidades de dados de forma supervisionada, não supervisionada ou por meio de reforço [8]. Neste projeto, o método abordado é o supervisionado, visto que ao montar o banco de dados de treinamento da rede, será informado à rede o “rótulo” de cada dígito.

O banco de dados foi criado com dígitos retirados de algumas placas, para que a rede neural aprenda o padrão de escrita e consiga responder de maneira mais eficaz. Foram fornecidas mil e oitenta imagens diferentes de cada dígito, ou seja, um banco de dados de dez mil e oitocentos exemplos. Quanto maior for o banco de dados de treinamento, mais robusta e eficaz a rede neural se torna. Cada dígito presente no banco possui as dimensões 30x50, ou seja, ao realizar o corte dos dígitos da placa, faz-se necessário o redimensionamento do dígito para as mesmas dimensões dos dados de treinamento, antes de ser classificado. A figura 6 ilustra alguns dos exemplos presentes no banco.

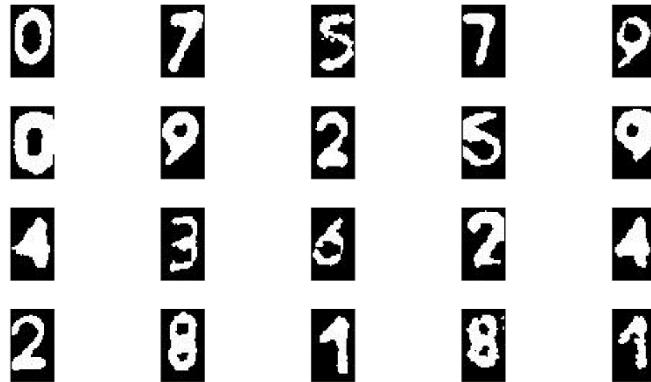


Fig. 6. Exemplo de imagens presentes no banco de dados

Outra característica importante que deve ser definida ao trabalhar com *Deep Learning*, é a arquitetura da rede neural. Para cada tipo de dado, aconselha-se um tipo de arquitetura, e no caso do reconhecimento de padrões em imagens, a arquitetura que atualmente proporciona melhores resultados é a rede neural convolucional, possuindo trabalhos realizados no reconhecimento de caracteres manuscritos [9], na detecção de pedestres [10] e faces [11]. Este tipo de rede neural foi construído se baseando no córtex visual de animais, tendo em vista que ao receber um estímulo visual na retina, a imagem é reconhecida devido a regiões do cérebro que identificam características específicas do sinal passado [12].

Como é mostrado na figura 7, camadas de convolução são aplicadas na imagem de entrada. Estas camadas são compostas de neurônios que estão conectados a um conjunto de pixels da imagem, tendo cada conexão um peso diferente. Os neurônios aplicam filtros em pedaços específicos da imagem, produzindo uma saída que é passada para a próxima camada, tendo em vista que na CNN, o neurônio se conecta apenas a um subconjunto de neurônios da camada anterior, abordagem diferente das redes neurais clássicas. O conjunto de neurônios de uma mesma camada é chamado de mapa. Após as camadas de convolução, é comum utilizar uma camada de agregação (*pooling*) que tem como função a redução das dimensões da imagem (altura e largura), enquanto a profundidade é aumentada, tendo assim, uma velocidade de treinamento maior.

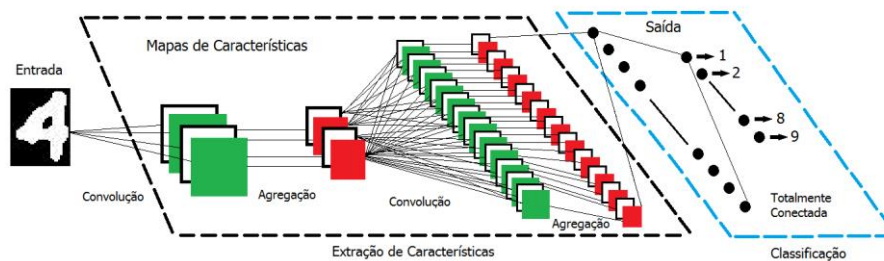


Fig. 7. Arquitetura de uma rede convolucional

Ao final das convoluções seguidas de agregações, é característico da arquitetura possuir uma camada de classificação visando rotular a resposta da rede se baseando nas saídas dos filtros. Utilizar uma camada totalmente conectada (*fully connected*) é uma maneira de menor custo computacional para aprender combinações não-lineares das características extraídas da imagem. Um exemplo da forma de trabalho da rede convolucional perante uma imagem é mostrado na figura 8.

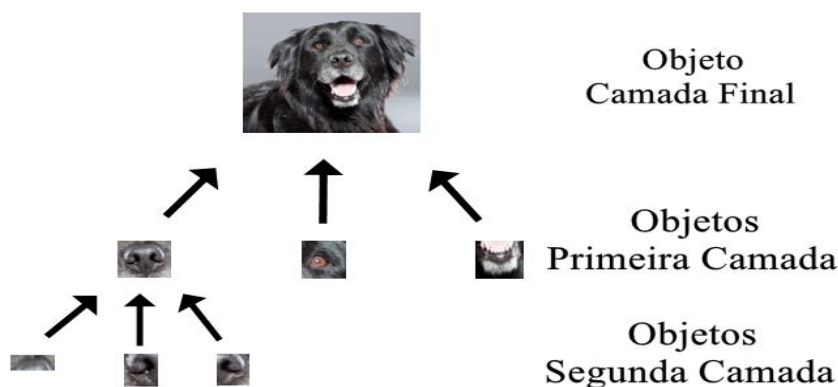


Fig. 8. Exemplo de aprendizagem de uma rede CNN

Para criar a rede neural, algumas informações precisam ser determinadas, como a taxa de aprendizado (*learning rate*), número de neurônios, quantidade de iterações, quantidade de camadas ocultas e a função de ativação nestas camadas, entre outros dados. Cada valor definido afetará diretamente o desempenho do classificador, logo, precisam ser escolhidos com um bom embasamento no assunto. As características da rede do presente trabalho foram definidas como representado na tabela 1, sendo válido citar a utilização de ReLU como função de ativação das camadas escondidas, por ser a função aconselhada para banco de dados supervisionados [13], e *Softmax*, que normaliza as saídas entre 0 e 1 em um problema multiclases, como função de ativação da saída.

Tabela 1. Camadas definidas da rede neural

1	Image Input	50x30x1 imagens com normalização <i>zero-center</i>
2	Convolution	20 5x5x1 convoluções com <i>stride</i> [1 1] e <i>padding</i> [0 0]
3	ReLU	ReLU
4	Average Pooling	2x2 average pooling com <i>stride</i> [2 2] e <i>padding</i> [0 0]
5	Convolution	20 5x5x1 convoluções com <i>stride</i> [1 1] e <i>padding</i> [0 0]
6	ReLU	ReLU
7	Average Pooling	2x2 average pooling com <i>stride</i> [2 2] e <i>padding</i> [0 0]
8	Fully Connected	2048 camadas totalmente conectadas
9	ReLU	ReLU
10	Dropout	50% dropout
11	Fully Connected	10 camadas totalmente conectadas
12	Softmax	Função de ativação <i>Softmax</i>
13	Classification Output	<i>Crossentropy</i> com '0', '1' e outras 8 classes

4 Resultados e Discussões

Primeiramente, a rede neural convolucional foi treinada com cinco imagens para cada dígito, totalizando cinquenta dados para o banco de treinamento, um número pequeno perante a grande capacidade do *Deep Learning*. Ao realizar o teste da rede, pôde ser visto que a eficiência da rede alcançava apenas 70%.

Foi construído então um banco de dados mais extenso e com maior variação nas imagens de cada algarismo. O banco contava com cem imagens para cada dígito, obtendo um total de mil dados de treinamento. A eficiência da rede saltou de 70% para 96,33%, alcançando um classificador mais robusto, porém com uma taxa de acertos ainda não satisfatória, tendo em vista resultados já obtidos com *Deep Learning*. Com isso, mais dados foram passados para o banco de treinamento, alcançando mil e oitenta imagens para cada dígito, ou seja, dez mil e oitocentos dados no total. A eficiência da

rede construída atingiu 99,78%, um valor satisfatório, se baseando em trabalhos de reconhecimento de dígitos [9], e um aumento significativo comparado aos testes anteriores que possuíam bancos de dados menores, o que era esperado por tratar-se de uma rede *Deep Learning*, pois quanto maior o banco de dados de treinamento, mais eficiente se torna o classificador. A tabela 2 mostra a matriz de confusão com os valores dados em porcentagem, tendo em vista que as linhas representam os rótulos conhecidos, enquanto as colunas representam os rótulos previstos.

Tabela 2. Matriz de confusão da rede neural criada

Dígito	0	1	2	3	4	5	6	7	8	9
0	100%	0%	0%	0%	0%	0%	0%	0%	0%	0%
1	0%	99%	0%	0%	0%	0%	0%	1%	0%	0%
2	0%	0%	100%	0%	0%	0%	0%	0%	0%	0%
3	0%	0%	0%	99%	0%	0%	0%	0%	1%	0%
4	0%	0%	0%	0%	100%	0%	0%	0%	0%	0%
5	0%	0%	0%	0%	0%	100%	0%	0%	0%	0%
6	0%	0%	0%	0%	0%	0%	100%	0%	0%	0%
7	0%	1%	0%	0%	0%	0%	0%	99%	0%	0%
8	0%	0%	0%	1%	0%	0%	0%	0%	99%	0%
9	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%

Um arquivo *guide* foi criado e confeccionado no intuito de dinamizar o projeto, tornando-o aplicável e fácil de ser verificado, podendo ser observado o resultado da binarização, o corte de cada dígito com as informações colhidas do 'BoundingBox' dos componentes e a resposta do classificador, conforme figuras 9 e 10.

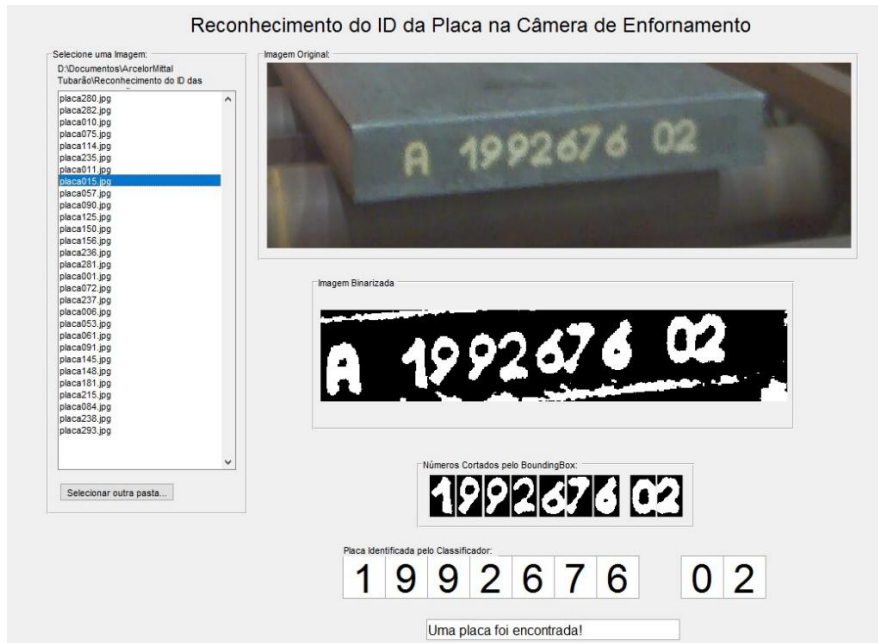


Fig. 9. Placa ilustrada anteriormente sendo reconhecida conforme o esperado

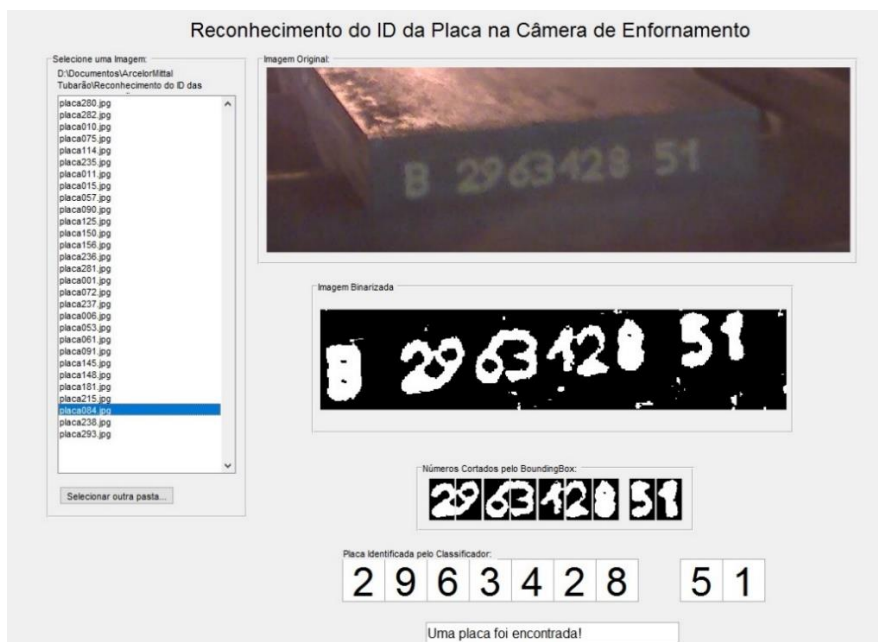


Fig. 10. Classificador reconhecendo a placa corretamente com uma imagem ruidosa

Baseando-se nas figuras 9 e 10, é possível notar a presença de ruídos nas imagens, mesmo com o ajuste da imagem realizado antes da binarização. Isto acontece pela falta de uniformidade nas imagens, sendo necessário assim, um pré-tratamento que satisfaça os diferentes tipos de iluminação do banco de imagens. Com isso, em algumas imagens os ruídos tornam-se mais frequentes do que em outras, porém, criando a lógica correta, os ruídos remanescentes conseguem ser ignorados, sendo reconhecidos apenas os dígitos do ID da placa.

O aumento significativo da taxa de acertos do classificador ao definir um banco de dados maior, permite confirmar a necessidade de possuir um banco de treinamento significativo. A eficiência alcançada de 99,78% superou metodologias tradicionais em reconhecimento de imagens, como a metodologia que utiliza o descritor HOG (Histograma de Gradientes Orientado) juntamente com um classificador SVM [14], que ao ser testada com o banco de dados construído, alcançou uma eficiência de 98%.

Em classificação de imagens a diferença de 1,78% é algo relativamente grande. No presente projeto, por exemplo, ao errar a classificação de um dos dígitos, todos os outros dígitos reconhecidos perdem a utilidade, tendo em vista que para o ID estar correto, todos os nove dígitos necessitam ser classificados corretamente.

5 Conclusão

Atualmente, o *Deep Learning* vem sendo utilizado como solução para diversos problemas de visão computacional, obtendo resultados significativos, com altas taxas de acertos, sendo considerado o estado da arte nesta área.

Visando uma forma de solucionar a baixa eficiência do atual sistema de reconhecimento do ID das placas presentes no LTQ de uma Siderurgia, o presente trabalho propôs um tratamento prévio da imagem, seguido de um novo classificador. Ao aplicar os filtros de mediana e gaussiano, as principais informações da placa não foram perdidas e os ruídos foram atenuados, tornando-se possível o corte dos dígitos do ID escrito na placa via codificação em Matlab, mostrando ser um ótimo caminho para vencer a falta de uniformidade das imagens colhidas pela câmera.

Com uma rede neural convolucional *Deep Learning* para extrair as características e classificar os dígitos, uma eficiência de 99,78% foi alcançada na rotulação das imagens de entrada, superando substancialmente o atual sistema, que possui uma taxa de 88% de acertos utilizando *Tesseract* para reconhecer e imprimir os dígitos, e técnicas clássicas de reconhecimento de padrões em imagens, como SVM com HOG, que ao ser testada com as imagens do projeto, alcançou 98% de eficiência.

Portanto, com a análise das respostas obtidas, o projeto mostrou ser uma alternativa viável para solucionar o problema ocorrente na Siderurgia, obtendo resultados satisfatórios. A metodologia utilizada possui potencial de ser aplicada e a mesma se encontra em estado de análise de implementação no processo real.

Referências

1. White, J. and Rohrer, G. Image thresholding for optical character recognition and other applications requiring character image extraction. *IBM J Res Dev.*, 27 (4): 400-411, 1983.
2. Wu, L., Yao, B., Peng, Z. and Guan, Y. An adaptive threshold algorithm for sensor fault based on the grey theory. *Advances in Mechanical Engineering*; 9(2): 1-7, 2017.
3. "Tesseract Open Source OCR Engine (main repository)," <https://github.com/tesseract-ocr>, 2017.
4. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A. "Going deeper with convolutions," in *CVPR 2015*, 2015.
5. Jain, R., Kasturi, R. and Schunck, G. B. *Machine Vision*. McGraw-Hill, first Edition, 1995.
6. Pedrini, H. *Apostila de Processamento de Imagens. Análise de Imagens Digitais: Princípios, Algoritmos e Aplicações*. Departamento de Informática-UFPR, 2004.
7. Krizhevsky, A., Sutskever, I. & Hinton, G. ImageNet classification with deep convolutional neural networks. In *Proc. Advances in Neural Information Processing Systems 25* 1090-1098 (2012).
8. Russell, S. and Norvig, P. *Artificial Intelligence: A Modern Approach*. [S.l.]: Prentice Hall, 3 ed., 2013.
9. LeCun, Y. Backpropagation applied to handwritten zip code recognition. *Neural Comput.*, 1(4):541-551, December 1989.
10. Sermanet, P., Kavukcuoglu, K., Chintala, S. and Lecun, Y. "Pedestrian detection with unsupervised multi-stage feature learning," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.
11. LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278-2324, November 1998.
12. Hubel, H. D. and Wiesel, T. N. "Receptive Fields of Single neurones in the Cat's Striate Cortex." *Journal of Physiology*. Vol 148, pp. 574-591, 1959.
13. Glorot, X., Bordes, A. and Bengio. Y. Deep sparse rectifier neural networks. In *Proc. 14th International Conference on Artificial Intelligence and Statistics*, 315-323, 2011.
14. Dalal, N. and Triggs, B. Histograms of oriented gradients for human detection, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 886-893, 2005.