Brazilian License Plate Character Recognition using Deep Learning

S. P. Peixoto, G. Cámara-Chávez, D. Menotti
Departamento de Computação
Universidade Federal de Ouro Preto
Ouro Preto, MG, Brazil
Email: {sirlenepg,guillermoc,menottid}@gmail.com

G. Gonçalves and W. R. Schwartz
Departamento de Ciência da Computação
Universidade Federal de Minas Gerais
Belo Horizonte, MG, Brazil
Email: {gabriel,william}@dcc.ufmg.br

***Abstract*—The quality of the acquired images is an important factor in the success of license plate recognition systems. Recognizing characters partially occluded by dirt, lighting, with varying sizes and colors is not a trivial task. We studied three different convolutional network approaches aimed at improving the features of these representations of character images through deep learning. We performed experiments in a dataset of Brazilian plates. The approach that learns filter weights through the back-propagation algorithm using data augmentation technique and the strategy of adding locally-connected layer to the Convolutional Network (CN) has obtained 99.83% accuracy for letters. The best accuracy found for the digits, 99.81%, was obtained by the approach that optimizes the architecture employing random filters.**

***Keywords*—Deep Learning; Convolutional Networks; Character Recognition; Vehicle License Plate.**

***Resumo*—A qualidade das imagens adquiridas é um fator importante para o sucesso de sistemas de reconhecimento de placas. Reconhecer caracteres parcialmente ocluídos pela sujeira, iluminação, com tamanhos e cores variados não é uma tarefa trivial. Neste trabalho, estudamos três abordagens de rede convolucional diferentes com o objetivo de melhorar as representações de característica destas imagens de caracteres através de *deep learning*. Nós realizamos experimentos em um dataset de placas brasileiras. A abordagem que aprende os pesos dos filtros através do algoritmo *back-propagation* utilizando a técnica *data augmentation* e a estratégia de agregação de camada localmente conectada à Rede Convolucional (Convolutional Network - CN) obteve 99.83% de precisão para letras. A melhor acurácia encontrada para os dígitos, 99.81%, foi obtida pela abordagem que otimiza a arquitetura empregando filtros aleatórios.**

I. Introdução

Um método de controle de acesso de veículos pode ser aplicado em sistemas eletrônicos de pagamento de pedágio e de controle de acesso à estacionamento, em sistemas de monitoramento de tráfego e de fiscalização, entre outras aplicações. Um típico sistema de reconhecimento de placas é composto por três etapas de processamento: detecção de placa, segmentação e reconhecimento dos caracteres. Recentemente, com o propósito de melhorar o desempenho desses sistemas, um grande número de métodos de reconhecimento de caracteres têm sido propostos, incluindo *template matching* [1], redes neuronais [2], [3] e *deep learning* [4], com redes convolucionais.

Durante a aquisição da imagem fatores como a diversidade de formatos da placa, a condição de iluminação, placa com caracteres apagados e até mesmo alguma sujeira na placa, podem causar a deturpação dos dados. Geralmente, o desempenho de métodos de aprendizagem de máquina está altamente relacionado com a capacidade do sistema de reconhecimento de extrair características que maximizam e minimizam similaridades intra e inter-classe, respectivamente.

Ao longo dos últimos anos, numerosos estudos mostram a eficácia das técnicas de *deep learning* em vários campos de aplicações desafiadoras. A técnicas de *deep learning* tem como objetivo mapear gradualmente a representação do pixel bruto para representações mais abstratas. Além do sucesso no reconhecimento de texto escrito à mão [5], técnicas de *deep learning* também foram aplicadas para os problemas de reconhecimento facial [6], reconhecimento de voz [7], reconhecimento de cena [8], entre outros.

Com o intuito de obter melhoria na representação das informações e consequentemente na tarefa de classificação, muitos trabalhos têm aplicado *deep learning* ao problema de reconhecer caracteres de imagens ruidosas [9], [10], [11], [12]. Em [9], os autores comparam e mostram a supremacia de dois descritores de aprendizagem em profundidade sobre dois descritores *hand-designed* usando classificadores otimizados na base de dados SVHN. Em [10], os autores incluíram na arquitetura de uma rede convolutional tradicional uma abordagem de aprendizado de características multi-estágio e diferentes métodos de *pooling* (*Lp pooling*) alcançando uma taxa de precisão igual de $95,1\%$ na base de dados SVHN (Street View House Numbers). Mais recentemente, em [11], foi proposta uma abordagem que integra as três etapas tradicionais de um sistema de reconhecimento (localização, segmentação e reconhecimento) usando uma rede convolucional e alcança uma precisão de $97,84\%$ no dataset SVHN.

Neste trabalho aplicamos diferentes técnicas/abordagens *deep learning* ao problema de reconhecimento de caracteres de placas brasileiras e investigamos e exploramos técnicas, tais como: 1) integração de camadas localmente conectadas à CN e 2) *data augmentation* para o treinamento. Também comparamos as abordagens *deep learning* com técnicas baseadas em descritores *hand-designed* utilizando imagens de caracteres de placas brasileiras organizadas em dois datasets: *Dígitos* e *Letras*. De acordo com nossos experimentos, observamos que a CN que otimiza a arquitetura empregando filtros aleatórios apresenta maior acurácia quando comparada com as outras abordagens avaliadas no dataset *Dígitos*. Já no dataset *Letras*, todas as abordagens *deep learning* obtiveram a mesma

acurácia. Os resultados também indicam que a aplicação das duas técnicas investigadas na abordagem que aprende os pesos dos filtros através de um algoritmo supervisionado, reduz a taxa de erro em ambos datasets.

O restante deste trabalho é organizado em três seções. Seção II detalha as abordagens *deep learning* avaliadas, a Seção III apresenta os experimentos e os resultados e, finalmente, a Seção IV conclui o trabalho e apresenta os possíveis trabalhos trabalhos.

II. ABORDAGENS *Deep Learning*

Uma CN é normalmente composta por camadas que geralmente executam uma série de quatro operações lineares e não-lineares sequenciais [13], tais como:

1) Convolução: calcula a intensidade de um dado pixel de acordo com a soma ponderada dos seus pixels vizinhos.
2) Ativação: mapeia a saída da operação de convolução com o objetivo de limitar o "neurônio" de saída.
3) *Spatial Pooling*: reduz a dimensão das características do mapa. Tem o efeito de reduzir a sensibilidade da saída do mapa a deslocamentos e outras formas de distorção, selecionando características invariantes que melhoram o desempenho de generalização.
4) Normalização: geralmente cria uma concorrência entre os mapas. Visa reforçar ainda mais a robustez da CN quanto ao vetor de características de saída.

Por meio dessas operações, a CN mapeia a imagem de entrada para uma representação de nível mais alto que pode ser vista como uma imagem multibanda, em outras palavras, a CN gradualmente transforma o pixel bruto em representações nas quais conceitos abstratos são destacados, esse processo é ilustrado na Figura 1. Esta representação multibanda é posteriormente aplicada a uma tarefa de classificação.

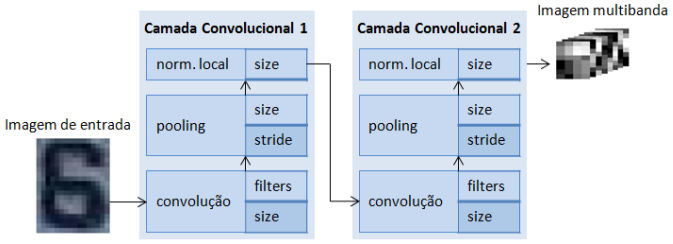

Figura 1. CN com duas camadas de convolução.

Neste trabalho, discutimos três CNs que aplicam diferentes abordagens para aprender/gerar os pesos dos filtros da operação de convolução. Enquanto a primeira abordagem otimiza a arquitetura empregando filtros aleatórios com média zero e norma igual a 1 [16], a segunda e a terceira utilizam um algoritmo supervisionado [14] e um não supervisionado [15], respectivamente, para aprender os pesos dos filtros. Ambas abordagens são brevemente descritas abaixo.

A. *Otimização de Arquitetura (OA) com Filtros Aleatórios*

Cada camada convolucional de uma CN é definida por n hiperparâmetros, tais como as dimensões dos filtro das operações e o número de filtros na operação de convolução, entre outros [13]. Considerando c camadas convolucionais e possíveis valores de cada um dos $n \times c$ hiperparâmetros (o *espaço de busca*), há uma imensidão de arquiteturas possíveis a serem avaliadas. Uma forma eficaz de explorar esse *espaço de busca* é através do emprego da abordagem OA como descrito em [16], em que algumas milhares de arquiteturas geradas aleatoriamente são avaliadas e a melhor topologia convolucional encontrada é retornada.

Os pesos dos filtros utilizados para avaliar as arquiteturas candidatas são gerados aleatoriamente a partir de uma distribuição uniforme $U(0,1)$ e normalizados para média zero e norma unitária, a fim de tornar a OA prática. A predição é realizada por máquinas de vetor de suporte (Support Vector Machine - SVMs) lineares. Mais detalhes sobre a abordagem OA podem ser encontrados em [16][1].

B. *Aprendizado Não Supervisionado de Filtros (ANSF)*

Nessa abordagem a CN é composta por apenas uma camada de convolução, sendo essa composta pelas quatro operações descritas no início dessa seção. Primeiramente é preciso encontrar o banco de filtros que serão aplicados na operação de convolução. Os filtros são aprendidos através de uma variante do algoritmo de aprendizado não supervisionado K-means (*Spherical K-means*) [15]. O *Spherical K-means* encontra o banco de filtros D de acordo com

$$\begin{aligned} \underset{D,s}{\text{minimize}} \quad & \sum_i ||Ds^i - x^i||_2^2 \\ \text{subject to} \quad & ||s^i||_0 \leq 1, \forall i \\ \text{and} \quad & ||D^j||_2 = 1, \forall j \end{aligned}$$

onde s^i, com $i = 1, ..., m$, é um "vetor de código"associado com a entrada x^i, e D^j, com $j = 1, ..., k$ é a coluna j do dicionário D, ou seja, é o j-ésimo filtro.

A entrada x^i corresponde a m sub-regiões pré-processadas de dimensão $p \times p$ extraídas aleatoriamente das imagens de dimensão $n \times n$ do conjunto de treinamento. O pré-processamento das sub-regiões tem o intuito de normalizar o brilho e o contraste e remover as correlações entre pixels vizinhos. Ele envolve duas operações: 1) normalização: para cada x^i subtrai-se a média das intensidades e divide-se pelo desvio padrão, e 2) *whitening* (também chamado de "*sphering*"): ZCA *whitening transform* (que difere da análise de componentes principais (PCA) apenas por uma rotação). O objetivo da equação é minimizar, sob duas restrições, o quadrado da diferença entre x^i e sua reconstrução correspondente Ds^i. A primeira restrição, $||s^i||_0 = 1$, significa que cada s^i é limitado a ter, no máximo, uma entrada diferente de zero. A segunda restrição exige que cada coluna do dicionário tenha comprimento unitário, impedindo-os de se tornar arbitrariamente grandes ou pequenos. D e s são otimizados de forma alternada. A solução ótima para s^i dado D é definida por $s_l^i = D^{l\intercal} x^i$, para $l = argmax_j D^{j\intercal} x^i$ e $s_j^i = 0$ para todos os outros $j \neq l$. Em seguida, mantendo-se s^i fixo, D é otimizado. Dado o banco de filtros aprendido D de dimensões $k \times p^2$, a CN inicia a extração de características das imagens do conjunto

[1] https://github.com/giovanichiachia/simple-hp

de treinamento. Por fim, o vetor de características extraído é submetido ao algoritmo de classificação supervisionado L2-SVM.

Optamos pela implementação disponível em[2] para avaliar a abordagem ANSF.

C. Aprendizado Supervisionado de Filtros (ASF)

Na abordagem ASF, a camada de convolução consiste em duas operações indispensáveis: convolução e ativação, e duas operações adicionais: normalização e *pooling*. A ordem destas duas últimas operações é intercambiável. A aprendizagem dos filtros das camadas de convolução, bem como os pesos das outras camadas é realizado de forma supervisionada através do algoritmo *back-propagation* que emprega a regra do gradiente descendente estocástico para atualizar os filtros e os seus pesos.

Embora a representação de características desta abordagem seja realizada pelas camadas convolucionais, o classificador consiste de uma (ou mais) camada completamente conectada com n valores de saída (em que n é o número de classes), e uma camada de regressão logística multinominal. A regressão logística multinomial é um método de classificação que generaliza a regressão logística para problemas multiclasse através da função softmax.

Em particular, nós usamos a biblioteca CUDA-convnet implementada em C++/CUDA por Alex Krizhevsky [14][3] para avaliar a abordagem ASF.

III. Resultados Experimentais

Nesta seção, nós descrevemos os conjuntos de dados avaliados e os detalhes dos experimentos, também apresentamos e analisamos os resultados.

A. Dataset *Placas Brasileiras*

Este *dataset* é composto por aproximadamente 2000 imagens de placas brasileiras (6000 letras e 8000 dígitos segmentados manualmente) e foi dividido em conjunto de treinamento (80% das amostras) e conjunto de teste (20% das amostras). As imagens foram extraídas de vídeos capturados na rua, a Figura 2 apresenta alguns exemplos. Baseados no típico modelo de placas brasileiro que é constituído por três letras seguidas de quatro dígitos, realizamos experimentos independentes nas categorias: dígitos e letras, ou seja separamos o *Dataset* Placas Brasileiras em dois conjuntos de dados, estes serão chamados de dataset Letras e dataset Dígitos a partir de agora. Observe que o tamanho das imagens em ambos conjuntos de dados é 32×32 pixels e que as mesmas foram convertidas em tons de cinza.

Figura 2. Exemplos do Dataset Placas Brasileiras

[2]http://www.cs.stanford.edu/ acoates/

[3]https://code.google.com/p/cuda-convnet/

B. Experimentos

Aqui nós avaliamos a eficácia das abordagens OA, ANSF e ASF descritas na Seção II para reconhecimento de caracteres de placas. Mostramos os experimentos aplicando essas abordagens nos conjuntos de dados Dígitos e Letras. Mais especificamente, apresentamos os resultados para a utilização de camadas localmente conectadas na CN e o uso de *data augmentation* no processo de aprendizagem. Nós também comparamos a eficácia das técnicas *deep learning* com outras abordagens baseadas em descritores *hand-designed*.

Tabela I. Configuração das arquiteturas que obtiveram os melhores resultados.

Operação	convolução		*pooling*		normalização
CN	filters	size	size	stride	size
OA-Dígitos	{64, 128}	{7, 3}	{3, 5}	{2, 1}	{-, 3}
OA-Letras	{192}	{7}	{7}	{2}	{-}
ASF	{64, 128, 192}	{3, 3, 3}	{3, 3, 3}	{2, 2, 2}	{9, 9, 9}

Otimização de Arquitetura: Devido ao espaço de memória necessário na abordagem OA para a representação das característica e para o aprendizado do SVM linear, foi necessário restringir o tamanho do conjunto de treinamento para ambos os conjuntos de dados, Dígitos e Letras, para 100 imagem/amostra por classe, de tal modo que as melhores arquiteturas convolucionais, ou seja, OA-Dígito e OA-Letras, respectivamente, puderam ser obtidos em dezenas de horas de processamento. Os valores dos hiperparâmetro da CNs OA-Dígito e OA-Letras são detalhados na Tabela I, em que o valor x^i na tupla $\{x^1, x^2, ..., x^n\}$ corresponde ao valor do hiperparâmetro da camada convolucional i e a ausência de qualquer operação na i-ésima camada de convolução é representada por '-'.

Nossos primeiros experimentos envolvem a avaliação do número de amostras utilizadas para treinar o SVM linear final, uma vez que a otimização de arquitetura foi realizada utilizando apenas 100 imagem/amostra por classe devido a restrições de memória. Estes resultados são apresentados na Tabela II. Observa-se que a taxa de erro diminui nos conjuntos de dados Dígitos e Letras à medida que o número de amostras de treinamento aumenta.

Tabela II. OA: variação do número de amostras do conjunto de treinamento.

CN	# treinamento	# teste	Taxa de Erro do Teste (%)
OA-Dígitos	1000	1589	3,27
OA-Dígitos	6387	1589	0,19
OA-Letras	2600	1197	1,50
OA-Letras	4833	1197	1,25

Aprendizado Não Supervisionado de Filtros: A Tabela III apresenta os resultados obtidos para a abordagem ANSF. Mais detalhadamente, este experimento consiste em avaliar a influência do número de filtros aplicados na camada de convolução no desempenho desta CN. Pode-se notar que em ambos datasets a acurácia aumenta conforme o número de filtros também aumenta.

Tabela III. ANSF: VARIAÇÃO DO NÚMERO DE FILTROS APLICADOS NA CAMADA DE CONVOLUÇÃO.

Dataset	# treinamento	# teste	# filtros	Taxa de Erro do Teste (%)
Dígitos	6387	1589	800	0,38
Dígitos	6387	1589	1600	0,44
Dígitos	6387	1589	3200	**0,31**
Letras	4833	1197	800	0,92
Letras	4833	1197	1600	1,67
Letras	4833	1197	3200	**1,25**

Aprendizado Supervisionado de Filtros: Com base nas arquiteturas convolucionais padrões publicamente disponíveis na biblioteca CUDA-convnet e no nosso domínio de conhecimento, nós definimos e avaliamos várias arquiteturas fundamentadas na abordagem ASF. Para treinar nossas CNs, seguimos a metodologia disponível na biblioteca CUDA-convnet[4]. Aquela que obteve os melhores desempenhos tanto no dataset Dígitos quanto no dataset Letras, ASF, está detalhada na Tabela I. As taxas de erro do teste obtidos são $3,08\%$ e $1,25\%$ para os datasets Dígitos e Letras, respectivamente.

Camadas Localmente Conectadas: A camada localmente conectada é similar a uma camada convolucional, mas a primeira não compartilha pesos como esta última faz. Ou seja, um conjunto diferente de filtros é aplicado a cada posição (x, y) do mapa/imagem de entrada, o que pode ajudar a descrever melhor as informações contidas nas imagens de caracteres, uma vez que diferentes regiões de um caractere têm diferentes estatísticas locais. Como a biblioteca CUDA-convnet suporta esse tipo de camada, investigamos a influência da inserção de uma, duas e três camadas localmente conectadas na CN após a última camada de convolução. A Tabela IV reporta as taxas de erro do teste obtidas, onde L corresponde ao número de camadas localmente conectadas adicionadas. Observe que as menores taxas de erro foram encontradas para $L = 1$ em ambos datasets. Daqui em diante, nós chamaremos essas arquiteturas de ASF-L-Dígitos e ASF-L-Letras.

Tabela IV. CAMADAS LOCALMENTE CONECTADAS.

Dataset	L	Taxa de Erro do Teste (%)
Dígitos	1	1,70
Dígitos	2	4,28
Dígitos	3	2,77
Letras	1	0,42
Letras	2	1,67
Letras	3	1,75

Data Augmentation: A técnica *data augmentation* combate *overfitting* através da ampliação artificial do conjunto de dados. Neste trabalho, o *data augmentation* consiste em gerar imagens transformadas a partir das imagens originais através de translações e reflexões horizontais. Seja I uma imagem de $n \times n$ pixels. A partir de i, sub-regiões quadradas (e seus reflexos horizontais) de ordem $n - 2 \times crop\_border$ são geradas, onde $2 \times crop\_border$ especifica a largura da região cortada em pixels. Investigamos a influência do *data augmentation* sobre a eficácia da ASF-L-Dígitos e ASF-L-Letras variando o parâmetro $crop\_border$. Estes resultados são apresentados na Tabela V. Como podemos observar, os melhores resultados para os dois conjuntos de dados foram obtidos para $crop\_border = 4$.

[4]https://code.google.com/p/cuda-convnet/wiki/Methodology

Tabela VI. COMPARAÇÃO COM DESCRITORES *hand-designed*

Dataset	Taxas de Erro do Teste (%)			
	Oblique Random Forest	Linear SVM	RBF SVM	Deep Learning
Dígitos	3,10	4,3	3,8	0,19
Letras	5,3	7,60	8,10	0,17

Tabela V. DATA AUGMENTATION.

Dataset	$crop\_border$	Taxa de Erro do Teste (%)
ASF-L-Dígitos	1	1,76
ASF-L-Dígitos	2	0,88
ASF-L-Dígitos	4	**0,44**
ASF-L-Letras	1	1,09
ASF-L-Letras	2	1,17
ASF-L-Letras	4	**0,17**

Comparação com outros Classificadores que utilizam Descritores Hand-designed*:* A fim de comparar as abordagens *deep learning* com outras abordagens baseadas em descritores *hand-designed*, realizamos experimentos em ambos conjuntos de dados usando o descritor *Histograms-of-Oriented-Gradients* (HOG) [17] com 8 bins e 4 tamanhos diferentes de células: 16×16, 8×8, 4×8 e 8×4. Todas as celulas têm *stride* configurado para 50%. Para a predição, usamos três classificadores. O primeiro é o método *Oblique Random Forest* (ORF) com um SVM em cada nó [18]. O ORF é uma abordagem que diferentemente do método *Random Forest*, utiliza um outro classificador para dividir os dados em cada nó de suas árvores. Os outros dois classificadores são SVM com kernel Linear e SVM com kernel *Radius Basis Function* (RBF). Os resultados são apresentados na Tabela VI. As menores taxas de erro são obtidas utilizando o classificador ORF, mas a sua acurácia está longe da obtida pela abordagem OA no dataset Dígitos e pela ASF no dataset Letras.

C. Breve Discussão

Os resultados mostram que as abordagens *deep learning* obtiveram acurácias muito superiores às que utilizam descritores *hand-designed* nos dois conjuntos de dados avaliados. A abordagem OA obteve a melhor acurácia no dataset Dígitos $99,81\%$, já no dataset Letras as três abordagens *deep learning* obtiveram a mesma taxa de erro $1,25\%$. A abordagem ASF reduziu significativamente esse erro de classificação no dataset Letras aplicando a estratégia que adiciona camadas localmente conectadas à CN e a técnica *data augmentation*.

Note que a abordagem OA apresentou outra vantagem sobre as demais: obteve a mesma acurácia no dataset Letras com apenas uma camada de convolução sem a operação de normalização e sem realizar pré-processamento nas imagens de entrada.

IV. CONCLUSÃO

Como uma importante conclusão do nosso estudo, observamos que a técnica *data augmentation* e a adição de camadas localmente conectadas aumentou significativamente as acurácias obtidas para ambos datasets na abordagem ASF. O resultado obtido no dataset Dígitos pela abordagem OA não foi alcançado pela abordagem ASF mesmo após a aplicação das técnicas investigadas. Isso sugere que a abordagem OA poderá obter resultados ainda melhores se combinada a tais

estratégias, essa hipótese será investigada em trabalhos futuros.

AGRADECIMENTOS

Os autores gostariam de agradecer à UFOP, CAPES e CNPq (projeto #307010/2014-7).

REFERÊNCIAS

[1] B. Li, B. Tian, Q. Yao, and K. Wang, "A vehicle license plate recognition system based on analysis of maximally stable extremal regions," in *Networking, Sensing and Control (ICNSC), 2012 9th IEEE International Conference on*. IEEE, 2012, pp. 399–404.

[2] J. Uddin, M. Rahman, M. Rakib, M. Ferdous, S. Kabir, M. Azam *et al.*, "Low complexity offline character recognition of license plate using perspective view," in *Computer and Information Technology (ICCIT), 2013 16th International Conference on*. IEEE, 2014, pp. 185–190.

[3] N. More and B. Tidke, "License plate identification using artificial neural network and wavelet transformed feature selection," in *Pervasive Computing (ICPC), 2015 International Conference on*. IEEE, 2015, pp. 1–5.

[4] D. Menotti, G. Chiachia, A. Falcao, and V. O. Neto, "Vehicle license plate recognition with random convolutional networks," in *Proceedings of the 2014 27th SIBGRAPI Conference on Graphics, Patterns and Images*. IEEE Computer Society, 2014, pp. 298–303.

[5] S. Vajda, Y. Rangoni, and H. Cecotti, "Semi-automatic ground truth generation using unsupervised clustering and limited manual labeling: Application to handwritten character recognition," *Pattern Recognition Letters*, vol. 58, pp. 23–28, 2015.

[6] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," *arXiv preprint arXiv:1503.03832*, 2015.

[7] X. Zhang, J. Trmal, D. Povey, and S. Khudanpur, "Improving deep neural network acoustic models using generalized maxout networks," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 215–219.

[8] Y. Yuan, L. Mou, and X. Lu, "Scene recognition by manifold regularized deep learning architecture," *IEEE Trans. on Neural Netw. Learn. Syst.*, pp. 1–12, 2015.

[9] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011, pp. 1–9.

[10] P. Sermanet, S. Chintala, and Y. LeCun, "Convolutional neural networks applied to house numbers digit classification," in *International Conference on Pattern Recognition (ICPR)*, 2012, pp. 3288–3291.

[11] Goodfellow, I. J. and Bulatov, Y. and Ibarz, J. and Arnoud, S. and Shet, V., "Multi-digit number recognition from street view imagery using deep convolutional neural networks," in *International Conference on Learning Representation*, 2014, arXiv preprint arXiv:1312.6082.

[12] J. Bai, Z. Chen, B. Feng, and B. Xu, "Image character recognition using deep convolutional neural network learned from different languages," in *Image Processing (ICIP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 2560–2564.

[13] N. Pinto, D. Doukhan, J. J. DiCarlo, and D. D. Cox, "A high-throughput screening approach to discovering good forms of biologically inspired visual representation," *PLoS computational biology*, vol. 5, no. 11, p. e1000579, 2009.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2012, pp. 1097–1105.

[15] A. Coates and A. Y. Ng, "Learning feature representations with k-means," in *Neural Networks: Tricks of the Trade*. Springer, 2012, pp. 561–580.

[16] D. Menotti, G. Chiachia, A. Pinto, W. R. Schwartz, H. Pedrini, A. X. Falcao, and A. Rocha, "Deep representations for iris, face, and fingerprint spoofing detection," *Information Forensics and Security, IEEE Transactions on*, vol. 10, no. 4, pp. 864–879, 2015.

[17] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2005, pp. 886–893.

[18] B. H. Menze, B. M. Kelm, D. N. Splitthoff, U. Koethe, and F. A. Hamprecht, "On oblique random forests," in *Machine Learning and Knowledge Discovery in Databases*. Springer, 2011, pp. 453–469.