RECONHECIMENTO DE COMPONENTES EM LINHAS FÉRREAS UTILIZANDO REDES NEURAIS CONVOLUCIONAIS

Luiz Augusto Zillmann da Silva^{*}, Vinicius Ferreira Vidal^{*}, Mathaus Ferreira da Silva^{*}, Murillo Ferreira dos Santos[†], Alexandre Lima de Carvalho^{*}, Augusto Santiago Cerqueira^{*}, Leonardo de Mello Honório^{*}

> * Universidade Federal de Juiz de Fora Faculdade de Engenharia Juiz de Fora, Minas Gerias, Brasil

[†]Centro Federal de Educação Tecnológica de Minas Gerais CEFET-MG Leopoldina, Minas Gerais, Brasil

Emails: luiz.zillmann@engenharia.ufjf.br, vinicius.vidal@engenharia.ufjf.br, mathaus.silva@engenharia.ufjf.br, murillo.ferreira@cefetmg.br, alexandre.carvalho@mrs.com.br, augusto.santiago@ufjf.edu.br, leonardo.honorio@ufjf.edu.br

Abstract— Due to the extensive proportions of Brazilian railways, there is a great demand for means to diagnose technical problems remotely and automatically. The railway can be recorded with a camera and possible defects detected using computer techniques, which helps human work and avoid damages. This work proposes a scene selection method using Deep Learning techniques, namely Convolutional Neural Networks (CNN), to recognize the poles, which gathers objects of interest to be inspected. Videos were obtained by driving through a railway, and the data divided and preprocessed for the network training and testing. A VGG network architecture served as start point, and after exhaustive search and comparisons of many techniques, two network patterns are presented and compared with practical test videos. The results yield more than 93% of success in training process for the classes, and time of execution in practical tests was considered satisfactory.

Keywords— Deep Learning, Convolutional Neural Networks, Image recognition, Pattern Recognition.

Resumo— Devido às proporções extensas da malha ferroviária no Brasil, existe demanda por meios de diagnosticar problemas técnicos de forma remota e automática. O trajeto percorrido pela malha pode ser capturado por uma câmera e possíveis defeitos detectados utilzando técnicas computacionais, oque contribui com o trabalho humano e evita danos aos componentes e terceiros. Este trabalho propõe uma solução para reconhecimento de características utilizando *Deep Learning*, especificamente Redes Neurais Convolucionais (RNC), para distinguir postes, os quais contêm vários componentes da rede de distribuição para serem inspecionados. Videos foram gravados ao longo das linhas, e os dados divididos e processados para treinamento e teste da rede. Uma rede com arquitetura VGG serviu como ponto de partida para treino e comparação, e após buscas e comparações exaustivas de diversas técnicas, dois padrões de rede são apresentados e comparados com testes práticos em videos. Os resultados retornam mais de 93 % de sucesso no processo de treinamento para das classes, e o tempo de execussão em testes práticos foi considerado satisfatório.

Palavras-chave Deep Learning, Redes Neurais Convolucionais, Reconhecimento de Imagem, Reconhecimento de Padrões.

1 Introdução

Redes ferroviárias são um importante meio de locomoção, seja para transporte de cargas ou passageiros. As mesmas possuem componentes elétricos em seu entorno, necessários para o seu funcionamento e para a distribuição de energia em geral, os quais necessitam de manutenção preventiva frequente (Chen et al., 2017). Essa é feita visualmente, com equipamentos de detecção acústica ou termográfica por um operário (LIMA FI-LHO, 2015), e em um trecho relativamente extenso de malha férrea há a possibilidade de fadiga visual, o que traria falhas na inspeção (Chen et al., 2018).

A inspeção automática das características da linha de distribuição vem a combater esse problema, com técnicas inteligentes para identificação dos elementos da mesma. A aquisição de dados é feita por câmeras, e as imagens e vídeos capturados analisados por técnicas computacionais para identificação das características desejadas.

Elementos da linha estão expostos, principalmente em postes, portanto sofrendo avarias do ambiente (LIMA FILHO, 2015). Elementos como cruzeta, isolador e chave fusível são exemplos de equipamentos que, danificados, podem causar curto circuitos e fugas de corrente, o que gera danos e perigo à linha e a terceiros (LUECKMANN, 2015).

Segundo (LeCun et al., 2015), técnicas convencionais de aprendizado de máquina eram limitadas quando era necessário o processamento de dados ditos "crus", demandando considerável trabalho e perícia do responsável pelo projeto do classificador durante a extração dos parâmetros. Devido a isso e maiores níveis de abstração alcançados por técnicas de *Deep Learning*, várias aplicações são vistas na literatura, como em (Chen and Jahanshahi, 2018), para detecção de fendas em usinas nucleares em combinação com descritor de Bayes ingênuo; em (Singh et al., 2017), para fusão de dados de sensores e mapeamento de pegadas; e (Silva et al., 2017), no reconhecimento de marcas em gado.

Este trabalho vem propor uma solução utilizando redes neurais convolucionais (RNCs) para identificação dos postes da rede de distribuição no entorno da malha férrea, a partir de dados obtidos em campo, uma vez que ali se encontram grande parte dos elementos críticos de inspeção. Essa técnica de *Deep Learning* se mostra entre as mais robustas para classificação de imagens, pois propõe reconhecimento de características mais profundas da imagem à medida que se evolui nas camadas da rede (Rosebrock, 2017).

Na literatura é possível encontrar trabalhos relacionados que propõem reconhecimento de elementos de forma automática. O trabalho de Faghih (Faghih-Roohi et al., 2016) foca sobre defeitos da malha ferroviária utilizando RNCs, enfatizando o uso da técnica sobre um banco de dados extenso de forma eficiente e robusta. Em (Chen et al., 2018), o monitoramento da qualidade dos suportes das catenárias na malha chinesa, que vem a conter bilhões de elementos a serem monitorados, é proposto com o emprego de RNCs modificadas por técnicas a fim de aprimorar a performance, concluindo com a comparação entre as redes propostas e demonstração de resultados sobre dados reais. Ambos os trabalhos focam sobre discriminação dos elementos em diversas classes para cada característica de interesse, enquanto este vem a reconhecer os postes para futuro processamento, como reconstrução 3D do ambiente, ou foco da câmera sobre a região de interesse.

O trabalho é organizado da seguinte forma: a Seção 2 descreve elementos de hardware e software utilizados no trabalho; a Seção 3 descreve como foi realizada a extração dos parâmetros para construção da rede; na Seção 4 estão detalhadas as etapas de construção, treinamento e conclusões sobre arquiteturas de rede propostas; a Seção 5 mostra o resultado da rede quanto ao reconhecimento de elementos em vídeos não utilizados durante a etapa de treinamento; e finalmente a Seção 6 traz as conclusões sobre as técnicas propostas.

2 Hardware e software utilizados

Para a aquisição das imagens foi utilizado um par de câmeras MAKO gigabit ethernet (Figura 1) com capacidade global shutter, capturando todos os pixeis da imagem no mesmo instante de tempo, no formato RGB, mesmo para objetos se movendo em altas velocidades. A aquisição é feita em forma de vídeo, processado como um conjunto de *frames*, separados em grupo para treinamento e teste. Cada *frame* representa uma cena de onde dados são extraídos para o algoritmo.



Figura 1: Câmera MAKO gigabit ethernet a esquerda, a direita suporte instalado para aquisição, abaixo o sistema montado.

Os resultados desse trabalho foram obtidos em uma máquina com processador i7 7th generation, sistema operacional Ubuntu 16.04, processamento paralelo CUDA-9.0 e placa de vídeo NVidia 1060.

3 Extração de parâmetros

Segundo (Rosebrock, 2017), as imagens de entrada da RNC devem ter dimensões iguais de largura e altura para aproveitar propriedades de álgebra matricial. Dimensões comuns são 32x32, 64x64, 96x96, 224x224, 227x227 e 229x229. Todavia, isso é um entrave para a aplicação em questão, pois o poste é uma característica bastante retangular e estendida em altura. Ao testar redes consagradas na literatura, como a própria Lenet (LeCun et al., 1998), com imagens retangulares, foi constatado que resultados semelhantes em termos de acurácia eram obtidos frente a imagens quadradas com a biblioteca Keras (Chollet, 2016), o que deu suporte para concluir que imagens retangulares poderiam ser classificadas no trabalho.

Uma primeira abordagem tentou extrair postes do conjunto de treinamento juntamente com uma parte considerável do fundo da imagem, como no exemplo da primeira imagem da Figura 2, com a intenção de localizar postes mesmo com o fundo em questão aleatório. Resultados de treinamento com menos de 70% de acurácia para ambas as classes e varredura de vídeos de teste mostraram que não seria uma abordagem confiável, chegando a, eventualmente, o fundo se tornar mais característico que o poste em si para detalhar a imagem, como visto na segunda imagem da Figura 2, causando também a ocorrência de falsos positivos. A solução para esse problema foi focar o máximo possível nas características do poste (terceira imagem da Figura 2), não absorvendo o fundo nas imagens do conjunto de treinamento. A partir disso, mesmo redes menos densas que as apresentadas na Seção 4 retornaram acurácia maior que 80% para ambas as classes.



Figura 2: A esquerda, poste prejudicado pelas características do fundo. Ao centro, poste prejudicado pelo céu, causando falsos positivos. A direita, poste em foco e ressaltado na imagem.

4 Construção da RNC

4.1 Conceito

Redes Neurais Convolucionais são modelos computacionais inspirados no sistema nervoso animal da mesma forma que as redes neurais artificias, utilizando, entretanto, camadas convolucionais em sua arquitetura. Fazem parte de um conjunto maior de técnicas de reconhecimento de padrões conhecida como *Deep Learning*, sendo constantemente aplicada a problemas de duas dimensões, como reconhecimento de imagens. Trata-se de uma técnica invariante à translação, rotação, escala ou mesmo iluminação (LeCun et al., 2015).

A Figura 3 ilustra uma arquitetura genérica de uma RNC para um problema de duas classes.



Figura 3: Arquitetura genérica de uma RNC.

A camada Convolucional dá nome a esse método de reconhecimento de padrões. Nesta camada ocorre a operação de convolução entre os valores de entrada, as intensidades dos pixeis, e os valores de pesos contidos nos *kernels*. O resultado obtido nesta operação é, então, aplicado a uma função de ativação, do tipo *Exponential Linear Units* (ELUs) para este trabalho (Clevert et al., 2015). Este conjunto contendo imagens de entrada, *kernels* e função de ativação é o neurônio deste tipo de rede. O aprendizado é realizado através da modificação dos pesos contidos nos *kernels*.

A Figura 4 a seguir ilustra um neurônio da rede, utilizando um *kernel* 3x3. Cada passo dado ao longo da imagem de entrada gera um resultado pertencente ao mapa de ativação. A aplicação de múltiplos *kernels* gera diferentes mapas, compondo um volume de saída.



Figura 4: Neurônio de uma RNC para kernels 3x3.

4.2 Arquitetura inicial

Para a identificação proposta por este trabalho, buscando obter o maior número de postes reconhecidos, a arquitetura de partida escolhida foi a mini-VGG (Rosebrock, 2017). As redes do tipo VGG possuem como características principais sua profundidade, ou seja, múltiplas camadas convolucionais, e a utilização de *kernels* 3x3 (Simonyan and Zisserman, 2014).

Além das camadas convolucionais, esta topologia possui também camadas dos tipos *Pooling*, utilizadas para redução do volume de saída, acelerando o treinamento (Masci et al., 2013); *batch normalization*, onde há a normalização das ligações entre camadas (Ioffe and Szegedy, 2015); e *Dropout*, onde há o sorteio de ligações a serem desfeitas para diminuir o *overfitting* (Srivastava et al., 2014).

Além disso, camadas totalmente conectadas e Softmax são utilizadas ao final da rede. A função é utilizada para cálculo da probabilidade de uma imagem pertencer a uma determinada classe, sendo empregada como classificador da rede (Krishnapuram et al., 2005). Sua entrada é mapeada para sua respectiva classe, através do produto $f(x_i, W) = Wx_i$, onde W é a matriz de pesos já contendo os valores de Bias e a função f é a função objetivo. A função f, também chamada de score ou pontuação em português, é simplificada como $f(x_i, W) = s$. A função Softmax, também chamada regressão logística multimodal, é definida como a probabilidade condicional normalizada de haver uma saída y_i dado uma entrada x_i (Rosebrock, 2017), como mostrado a seguir.

$$P(Y = y_i, X = x_i) = \frac{e^{s_{y_i}}}{\sum_j e^{s_j}}$$
(1)

A função Softmax também foi utilizada para

cálculo de perdas através da função cross-entropy loss (Nasr et al., 2002), ou função de perda por entropia cruzada, obtida, para cada entrada, aplicando-se o logaritmo natural negativo à Softmax.

$$L_i = -ln(\frac{e^{s_{y_i}}}{\sum_j e^{s_j}}) \tag{2}$$

A função de perda sobre todas as entradas pode ser calculada através da média das perdas para cada entrada.

$$L = \frac{1}{N} \sum_{i}^{N} L_{i} \tag{3}$$

Onde L_i é o resultado obtido na Equação 2 e N é o número total de imagens para entrada.

4.3 Treinamento

Para treinamento supervisionado, foi definido um conjunto contendo imagens de postes, ditas positivas, e imagens contendo quaisquer elementos a exceção de postes, ditas negativas. Este conjunto de treinamento contém 200 negativas e 80 positivas, totalizando 280 imagens no conjunto de treinamento. Neste conjunto, a divisão foi realizada seguindo a proporção 75% para treinamento e 25% para validação.

A abordagem escolhida foi a minimização da função de perda, através da aplicação do gradiente descendente estocástico, utilizando momento para aumentar a atuação do gradiente se este se mantiver na mesma direção na iteração seguinte, ou diminuir a atuação caso haja mudança de direção. As Equações 4 e 5 demonstram a atualização dos pesos com aplicação de momento (Qian, 1999)(Ruder, 2016).

$$\mathbf{W} = \mathbf{W} + \mathbf{V} \tag{4}$$

Sendo,

$$\mathbf{V} = \gamma \mathbf{V} - \alpha \nabla_W f(\mathbf{W}) \tag{5}$$

Onde **W** representa o vetor de pesos, γ é o coeficiente do momento V, normalmente um valor entre $0,5 \in 0,9, \alpha$ é a taxa de aprendizado e $\nabla_W f(\mathbf{W})$ é o gradiente da função a ser minimizada.

Também foi utilizada para treinamento a aceleração de Nesterov (Ruder, 2016), que pode ser considerada como uma correção para o momento. A atualização é feita através da modificação na Equação 5, de forma a calcular não o gradiente para W atual, mas para estimar a posição futura, resultando na Equação 6.

$$\mathbf{V} = \gamma \mathbf{V} - \alpha \nabla_W f(\mathbf{W} - \gamma \mathbf{V}) \tag{6}$$

4.4 Topologias propostas

Além da mini-VGG, duas topologias alternativas propostas por este trabalho foram treinadas, produzidas a partir de testes e mudanças realizados na topologia anterior. A Figura 5 ilustra resumidamente a arquitetura chamada por este trabalho de Rede Rápida, mostrando os vetores em cada camada desde a entrada RGB até a última camada onde é realizada a classificação. Da mesma forma, a Figura 6 representa a Rede Profunda. As setas coloridas de azul representam a aplicação de Dropout. O objetivo da rede profunda é a obtenção de uma menor quantidade de falsos positivos, ainda que tenha um maior custo computacional. Já a rápida, como o próprio nome sugere, tem foco na redução desse custo afetando o menos possível a acurácia do sistema.



Figura 5: Rede Rápida.



Figura 6: Rede Profunda.

Para atingir as arquiteturas propostas, foram realizadas e reportadas diversas modificações como número de camadas convolucionais, variação de volumes de ativação das camadas convolucionais, dimensões variáveis para campo de recebimento das camadas (kernel), e inserção de técnicas como dropout e batch normalization.

4.5 Resultados de Treinamento

A Figura 7 mostra o resultado em um treinamento da rede mini-VGG. Apesar de mostrarem acurácia comparável às propostas (acima de 90%), as perdas na validação foram maiores, tendo sua curva oscilado em torno de 0,2 ao longo das épocas de treinamento.

Em contrapartida, o resultado de treinamento para a rede profunda proposta, incrementada com *batch normalization* após as duas primeiras camadas convolucionais (ilustrada pela Figura 8), apresenta um comportamento inesperado inicialmente segundo a literatura (Rosebrock, 2017), para os valores de perda na validação, mesmo atingindo um resultado satisfatório posteriormente.

Já na Figura 9, a função de perda na validação apresenta um decaimento suave, característicos da aplicação de momento e aceleração de Nesterov, obtendo valores semelhantes aos apresentados pelas arquiteturas anteriores, atingindo 0,2. Os valores de acurácia evoluem da mesma forma, seguindo assintoticamente para 100%. Devido ao comportamento das curvas e aos resultados de perda na validação, são adotadas por este trabalho as arquiteturas rápida e profunda sem camada de *batch normalization*.



Figura 7: Evolução da rede mini-VGG com dois agrupamentos de camadas convolucionalbatchnormalization-dropout.



Figura 8: Evolução da rede profunda com camada de *batch normalization*.



Figura 9: Evolução do treinamento para rede profunda.

A Tabela 1 mostra a média sobre as porcentagens de acurácia para conjunto de validação sobre as classes postes (positivas) e não postes (negativas), para as redes avaliadas. Nela é observada a superioridade das topologias propostas, principalmente com relação as imagens positivas. Comparando as redes rápida e profunda, como esperado a segunda mostrou maior acerto durante a validação, com um custo computacional maior devido a quantidade de camadas convolucionais. Mais comparações entre as redes são realizadas na Seção 6.

Tabela 1: Acurácias médias para conjunto de validação.

Rede	Positivos	Negativos	Média
mini-VGG	0,47	0,80	$0,\!64$
Rápida	0,93	0,99	0,96
Profunda	0,97	0,99	0,98

5 Resultados práticos

Para avaliação do desempenho da arquitetura escolhida, foi realizado o deslocamento de uma janela retangular ao longo de cada frame de um vídeo (Figura 10), variando o tamanho da janela ao final de cada varredura, fazendo com que os pixeis contidos nesta janela fossem avaliados pela RNC. É importante ressaltar que a imagem obtida na janela é pré-processada para estar em conformidade com as dimensões da entrada da RNC. O resultado da função Softmax na classe poste para cada janela é comparado a um threshold ajustado empiricamente, sendo o maior resultado considerado poste no frame em questão. São contabilizados o valor da pontuação obtida pela janela e o tempo de processamento para aquela imagem. A Figura 11 ilustra um resultado para um frame aleatório.



Figura 10: Janela varrendo um frame aleatório.

Em cada vídeo coletado para a avaliação e para cada arquitetura, foram avaliados tempo de processamento para cada *frame*, postes perdidos, postes encontrados (chamados positivos) e falsos positivos (quando não postes são marcados como postes). Foram então calculados a média e desvio padrão das pontuações de acerto para cada vídeo.



Figura 11: Poste encontrado durante varredura em vídeo de teste, com tempo em centésimos de segundo.

Os resultados são apresentados nas Tabelas 2 e 3, para o vídeo 1, de 27 postes e 994 frames; e 4 e 5, para o vídeo 2, de 10 postes e 1105 *frames*. É importante frisar que um mesmo poste físico aparece em diversos *frames* dos vídeos.

Tabela 2: Resultados de postes perdidos, frames positivos e falsos positivos para Vídeo 1.

Rede	Postes	Positivos	Falsos
	Perdidos		Positivos
mini-VGG	12	70	805
Rápida	4	166	73
Profunda	4	204	194

Tabela 3: Resultados de tempo de processamento, pontuação média de saída e desvio padrão para Vídeo 1.

Dodo	Tempo	Média da	Desvio
neue	(s)	Saída	Padrão
mini-VGG	0,25	0,951	0,021
Rápida	0,10	0,955	0,026
Profunda	0,15	0,977	0,027

Tabela 4: Resultados de postes perdidos, frames positivos e falsos positivos para Vídeo 2.

Rede	Postes Perdidos	Positivos	Falsos Positivos
mini-VGG	1	161	783
Rápida	1	191	24
Profunda	0	268	41

Tabela 5: Resultados de tempo de processamento, pontuação média de saída e desvio padrão para Vídeo 2.

Rede	Tempo (s)	Média da Saída	Desvio Padrão
mini-VGG	0,25	0,949	0,021
Rápida	0,10	0,969	0,027
Profunda	0,15	0,971	0,026

Como pode ser observado nas Tabelas 3 e 5, o tempo das redes propostas é reduzido em até mais de 50% (para a rede rápida) em relação ao ponto de partida da rede mini-VGG, mesmo apresentando um número maior de camadas convolucionais. Isso se deve à ausência de batch normalization entre as camadas das RNCs propostas, o qual reduziu não só o tempo como também aumentou a taxa de acertos para o banco de dados em questão. Em uma comparação entre as duas arquiteturas propostas, o tempo da rede rápida chega a dois terços do mesmo para a rede profunda, o que é relevante para aplicações online. As médias dos valores da função Softmax encontrados para ambas as redes propostas são superiores à mini-VGG, o que demonstra identificação mais robusta dos postes de forma geral.

Em relação à quantidade de postes perdidos e falsos positivos (Tabelas 2 e 4) a superioridade de ambas as redes propostas em relação à rede mini-VGG é clara, e a rede profunda obtém mais frames com postes reconhecidos (aproximadamente 23 e 40% a mais para vídeos 1 e 2, respectivamente), o que auxilia a identificação de falhas e problemas em relação aos mesmos. Em contrapartida, o número de postes perdidos para as duas redes propostas é o mesmo, o que em conjunto com o custo computacional reduzido favorece a utilização da rede rápida. Além disso, resultado de falsos positivos para a rede profunda contesta a hipótese de que aumentando o número de camadas convolucionais teríamos menor quantidade de falsos positivos, sugerindo a possibilidade de overfitting.

Dois vídeos de comparação entre as redes para um mesmo trecho de filmagem estão disponíveis no Youtube, onde o desempenho da rede profunda pode ser visto com o link https://www.youtube.com/watch?v=d9qV2A0cY fo&feature=youtu.be, para е rede rápida a com 0 link https://www.youtube.com/watch?v=HqSE7WvH _MY&feature=youtu.be.

6 Conclusões

O trabalho apresenta um solução para o reconhecimento automático de postes no entorno de uma linha férrea utilizando a técnica de redes neurais convolucionais, com o objetivo de auxiliar no monitoramento da mesma. São propostas duas arquiteturas para a rede, e a comparação entre os resultados em situações práticas é demonstrada. Ambas as redes propostas apresentam taxas acima de 93% de acerto para classes de poste e negativos no treinamento, e resultados práticos mostram semelhança quanto a postes físicos encontrados. Mesmo com a superioridade da rede profunda em termos de postes encontrados em *frames*, o tempo de execução médio da rede rápida e a menor quantidade de falsos positivos a tornam mais vantajosa para aplicações *online*.

É importante ressaltar que, com a técnica usada e uma quantidade relativamente pequena de banco de dados para treinamento (frente à extensão da linha), a característica do poste foi bem identificada, conforme Seção 5, mesmo frente a *backgrounds* variados em diversos *frames*, o que traz um bom resultado em trechos diversos da malha.

Como trabalho futuro, outros elementos de interesse na linha podem ser incluídos na identificação, principalmente relacionados aos postes identificados a princípio. Outras técnicas podem ser programadas em conjunto e a partir da identificação por *Deep Learning*, como a reconstrução em três dimensões do ambiente, com foco nos objetos identificados.

Referências

- Chen, F.-C. and Jahanshahi, M. R. (2018). Nbcnn: Deep learning-based crack detection using convolutional neural network and naïve bayes data fusion, *IEEE Transactions on In*dustrial Electronics 65(5): 4392–4400.
- Chen, J., Liu, Z., Wang, H. and Liu, K. (2017). High-speed railway catenary components detection using the cascaded convolutional neural networks, *Imaging Systems and Techni*ques (IST), 2017 IEEE International Conference on, IEEE, pp. 1–6.
- Chen, J., Liu, Z., Wang, H., Núñez, A. and Han, Z. (2018). Automatic defect detection of fasteners on the catenary support device using deep convolutional neural network, *IEEE Transactions on Instrumentation and Measurement* 67(2): 257–269.
- Chollet, F. (2016). Keras documentation.
- Clevert, D.-A., Unterthiner, T. and Hochreiter, S. (2015). Fast and accurate deep network learning by exponential linear units (elus), arXiv preprint arXiv:1511.07289.
- Faghih-Roohi, S., Hajizadeh, S., Núñez, A., Babuska, R. and De Schutter, B. (2016). Deep convolutional neural networks for detection of rail surface defects, *Neural Networks*

(IJCNN), 2016 International Joint Conference on, IEEE, pp. 2584–2589.

- Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift, *International conference on machine learning*, pp. 448– 456.
- Krishnapuram, B., Carin, L., Figueiredo, M. A. and Hartemink, A. J. (2005). Sparse multinomial logistic regression: Fast algorithms and generalization bounds, *IEEE transacti*ons on pattern analysis and machine intelligence 27(6): 957–968.
- LeCun, Y., Bengio, Y. and Hinton, G. (2015). Deep learning, *nature* **521**(7553): 436.
- LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P. (1998). Gradient-based learning applied to document recognition, *Proceedings of the IEEE* 86(11): 2278–2324.
- LIMA FILHO, A. D. S. (2015). Manutenção em redes de distribuição de energia elétrica, *REPOSITÓRIO DE RELATÓRIOS-Engenharia Elétrica* (1).
- LUECKMANN, C. (2015). Inspeção em linhas de distribuição de energia elétrica lages, REPOSITÓRIO DE RELATÓRIOS-Engenharia Elétrica (1).
- Masci, J., Giusti, A., Ciresan, D., Fricout, G. and Schmidhuber, J. (2013). A fast learning algorithm for image segmentation with maxpooling convolutional networks, *Image Pro*cessing (ICIP), 2013 20th IEEE International Conference on, IEEE, pp. 2713–2717.
- Nasr, G. E., Badr, E. and Joun, C. (2002). Cross entropy error function in neural networks: Forecasting gasoline demand., *FLAIRS Conference*, pp. 381–384.
- Qian, N. (1999). On the momentum term in gradient descent learning algorithms, *Neural* networks **12**(1): 145–151.
- Rosebrock, A. (2017). Deep Learning for Computer Vision with Python, PYImageSearch.
- Ruder, S. (2016). An overview of gradient descent optimization algorithms, *arXiv preprint arXiv:1609.04747*.
- Silva, C., Welfer, D., Gioda, F. P. and Dornelles, C. (2017). Cattle brand recognition using convolutional neural network and support vector machines, *IEEE Latin America Transactions* 15(2): 310–316.

- Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition, arXiv preprint ar-Xiv:1409.1556.
- Singh, M. S., Pondenkandath, V., Zhou, B., Lukowicz, P. and Liwickit, M. (2017). Transforming sensor data to the image domain for deep learning - an application to footstep detection, Neural Networks (IJCNN), 2017 International Joint Conference on, IEEE, pp. 2665–2672.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting, *The Journal* of Machine Learning Research 15(1): 1929– 1958.