

**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA
FACULDADE DE TECNOLOGIA DE BOTUCATU
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS**

MATHEUS HENRIQUE BALDI DE ALMEIDA

**DESEMPENHO DA DEEP LEARNING NA ANÁLISE E CATEGORIZAÇÃO DE
IMAGENS DE DEFEITO DE MADEIRA**

Botucatu-SP
Janeiro – 2017

**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA
FACULDADE DE TECNOLOGIA DE BOTUCATU
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS**

MATHEUS HENRIQUE BALDI DE ALMEIDA

**DESEMPENHO DA DEEP LEARNING NA ANÁLISE E CATEGORIZAÇÃO DE
IMAGENS DE DEFEITO DE MADEIRA**

Orientador: Prof. Me. Roger Cristhian Gomes

Co-orientador: Prof. Dr. Osvaldo César Pinheiro de Almeida

Artigo Científico apresentado à FATEC -
Faculdade de Tecnologia de Botucatu, como
exigência para cumprimento do Trabalho de
Conclusão de Curso no Curso Superior de
Análise e Desenvolvimento de Sistemas

Botucatu-SP
Janeiro – 2017

DESEMPENHO DA DEEP LEARNING NA ANÁLISE E CATEGORIZAÇÃO DE IMAGENS DE DEFEITO DE MADEIRA

Matheus Henrique Baldi de Almeida¹ Roger Cristhian Gomes² Osvaldo César
Pinheiro de Almeida³

RESUMO

A inteligência artificial, no que diz respeito ao campo de pesquisa, tem tido grandes avanços e contribuído bastante em várias áreas, tal como a análise e categorização de imagens com o aprendizado de máquinas. Existem várias técnicas específicas para o reconhecimento de imagens e sua categorização, e uma delas utiliza redes neurais artificiais, que envolve o estudo específico, a extração das características por meio da análise de dados da imagem do objeto que está sendo analisado e a especificação de qual será o impacto delas no modelo neural para cada uma das categorias, o que muitas vezes envolve a imersão do pesquisador numa área ou campo de pesquisa que não é o dele. A *deep learning*, utilizando-se das redes neurais convolucionais artificiais tem a capacidade de aprender e extrair as características durante o treinamento, sem a especificação dessas características no modelo, sendo que, geralmente, apresentam até mesmo resultados melhores do que aqueles observados por modelos de redes neurais que tiveram as características observadas programadas por humanos. O objetivo desse trabalho é aplicar a *deep learning* em um conjunto de imagens de tábuas de madeira, avaliando seu desempenho na categorização dessas imagens. Utilizando uma ferramenta chamada *Keras*, uma biblioteca *python* que auxilia no carregamento de conjuntos de dados, produção e avaliação de modelos de redes convolucionais, foram elaborados alguns modelos e avaliados seus desempenhos na análise e categorização de imagens de defeito de madeira, obtendo resultados promissores, sendo o melhor deles um percentual de erro na categorização de 5,44%.

Palavras-Chave: Análise de dados. Aprendizado de Máquinas. Redes Convolucionais Artificiais. *Keras*.

¹ Graduando em Análise e Desenvolvimento de Sistemas, FATEC, Av. José Ítalo Bacchi, s/n – Jd. Aeroporto – Botucatu/SP – CEP: 18606-851. E-mail: matheushbaldi@gmail.com

² Professor Mestre, FATEC - Av. José Ítalo Bacchi, s/n – Jd. Aeroporto – Botucatu/SP – CEP: 18606-851. E-mail: rgomes@fatecbt.edu.br

³ Professor Doutor, FATEC - Av. José Ítalo Bacchi, s/n – Jd. Aeroporto – Botucatu/SP – CEP: 18606-851. E-mail: cesar@fatecbt.edu.br

DEEP LEARNING PERFORMANCE IN WOOD DEFECT IMAGES ANALYSIS AND CATEGORIZATION

ABSTRACT

Artificial intelligence, in its research field, is getting better and contributing a lot in many different areas, one example is image analysis and categorization with machine learning. There are many different and specific techniques for image recognition and its categorization, one of them apply artificial neural networks, which involves a specific study, the features extraction by data analysis of the object and the specifications of how it will impact in the neural model for each of the defined categories. Usually it takes the researcher to immerse in an area or research field which is not related with his own area. The Deep Learning, making use of artificial convolutional neural networks is capable of learning and extract the features during the training, without the specification of its features in the model, and usually even getting better results than those ones obtained from neural networks models that has the observed features implemented by humans. The objective of this paper is to apply the deep learning in a wood defect image dataset, noticing its performance in the categorization of these images. Using a tool named Keras, which is a python library that helps the loading of datasets, development and evaluation of convolutional networks models, we made some models and evaluated its performance in the analysis and categorization of a wood defect images dataset, getting promising results, the best of it was 5.44% in the baseline categorization error.

Keywords: Artificial Convolutional Neural Networks. Features extraction. Machine Learning. Keras.

1 INTRODUÇÃO

A inteligência artificial tem sido tema cada vez mais recorrente no desenvolvimento de ferramentas que tanto auxiliam na tomada de decisões quanto são capazes, de certa forma, de tomar decisões de forma autônoma com base nos dados de entrada que recebe. Estudos têm recebido grandes avanços em várias áreas de aplicação que vão muito além da informática, graças aos padrões que as máquinas conseguem reconhecer em imagens e outras fontes.

A análise de imagens é um campo bastante explorado nesse quesito. Existem muitos *softwares* que são desenvolvidos e feitos cuidadosamente de forma a tratar a imagem e então aplicar algoritmos que façam a extração dos dados interessantes contidos na imagem. No entanto, nesse caso, os computadores são instruídos a seguirem uma sequência de ações que utilizarão categorias pré-definidas na programação, ou seja, deve ser feito um estudo minucioso para encontrar quais são as características relevantes a serem exploradas e utilizadas no processamento.

Nesse ponto surge o aprendizado de máquina, que tem como objetivo fazer com que os computadores sejam programados a aprender como processar os dados que estão sendo inseridos (SHALEV-SHWARTZ; BEN-DAVID, 2014), ou seja, em vez de os passos serem previamente programados, os computadores aprendem a extrair características e usá-las para determinado fim, tal como a análise e categorização de imagens.

Existem algumas formas de se desenvolver programas capazes de aprender sozinhos, uma delas é por meio da utilização de redes neurais. Redes neurais artificiais são modelos computacionais compostos de unidades de processamento adaptáveis e com conexões densas que fornecem uma grande variedade de soluções para problemas, tais como classificação de padrões, síntese e análise de padrão de vocal, compreensão de imagens, previsões e outros (HASSOUN, 1995). Essa capacidade de auto aprendizado se dá devido à característica de as redes neurais artificiais serem modelos simplificados que imitam o funcionamento de uma estrutura de rede neural biológica (YEGNANARAYANA, 2009).

Entretanto, recentemente uma nova abordagem relacionada com aprendizado de máquina e redes neurais tem ganhado muita força, é a chamada *Deep Learning*. *Deep learning* pode ser entendido como uma classe de técnicas de aprendizado de máquina que explora muitas camadas de informação não lineares. Basicamente, *deep learning* trata do aprendizado de múltiplos níveis de representação e abstração que ajudam a construir

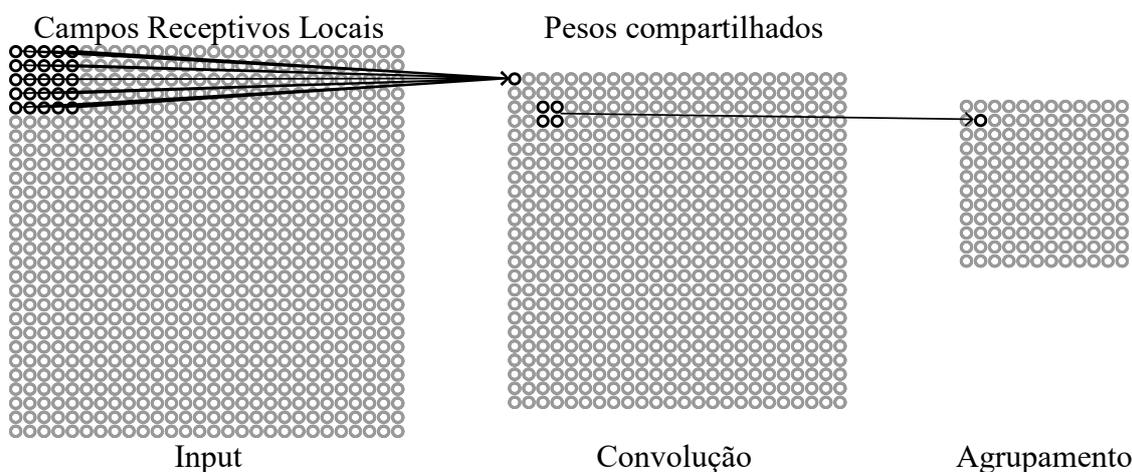
sentido em dados, tais como imagens, sons e texto. Em suas diversas definições, duas características estão sempre presentes e permitem a representação sucessivamente mais alta de camadas mais abstratas, sendo um modelo que consiste de múltiplas camadas de processamento de informação não linear e também um método de aprendizado supervisionado, onde os dados inseridos já estão classificados e categorizados, e não supervisionado, onde a máquina classifica por conta própria os dados com base nos padrões encontrados (DENG; YU, 2014).

Porém o treinamento de uma rede neural utilizando *deep learning* pode ser bastante difícil e complexo, por isso utiliza-se de um outro conceito importantíssimo chamado de rede neural convolucional artificial, ou simplesmente rede convolucional. As redes convolucionais têm uma diferença fundamental em relação as redes neurais comuns, os neurônios não utilizam uma ligação todos-todos de uma camada para outra, ou seja, em vez de se ligar cada um dos neurônios da primeira camada com todos os neurônios da segunda camada, é aplicada uma outra técnica que é chamada de campos receptivos locais. Essa técnica, conforme ilustrado na Figura 1, consiste em distribuir os neurônios numa matriz, assemelhando-se à imagem que será usada como *input*. Cada neurônio receberá o valor de um *pixel*. Tendo como exemplo uma imagem de 28x28 *pixels*, a técnica consistiria em selecionar um determinado campo dessa imagem com um tamanho específico, por exemplo um campo de 5x5. Dessa forma, esse campo de 5x5 teria seus valores filtrados para um único neurônio da próxima camada. Essa filtragem é chamada de convolução. No momento em que essa imagem é convolucionada são gerados mapas de característica, ou *feature maps*. Cada um desses mapas é responsável pela especificação de uma característica. No caso da entrada de 28x28 *pixels*, tendo um campo de 5x5 e um passo de 1 neurônio, haveriam mapas de características na camada de convolução com o tamanho de 24x24 neurônios. Todos os neurônios de um determinado mapa de características receberão um mesmo peso e valor de passagem, esse é outro conceito importante das redes convolucionais e é chamado de pesos compartilhados. Ele é quem permite que cada mapa analise uma característica específica. (NIELSEN, 2015).

Outro conceito importante é o de agrupamento, ou *pooling*, ilustrado na Figura 1. Geralmente, após uma camada de convolução, a próxima camada do modelo de rede convolucional será uma camada de agrupamento. O objetivo dessa camada é gerar uma saída simplificada das características capturadas pela camada de convolução. Os mapas de características são dispostos numa matriz, assim como os dados de entrada, dessa forma, no processo de agrupamento, geralmente usa-se um campo de 2x2 desses mapas e o valor desses campos é generalizado para um único neurônio da camada de

agrupamento. Essa generalização pode ser feita de várias formas, porém a mais comum é conhecida como *max-pooling* que consiste em utilizar o valor mais significativo do campo 2x2 e atribuir ao neurônio da camada de agrupamento. Diferentemente da convolução, os campos de agrupamento não repetem neurônios já agrupados, dando a ideia de que o passo para agrupamento seria igual ao tamanho do campo, no caso do exemplo seria 2. Isso significa que a camada de agrupamento para o modelo sugerido teria 12x12 neurônios, levando em conta que a camada de convolução apresentou 24x24 *pixels* (NIELSEN, 2015).

Figura 1. Campos receptivos locais, convolução e agrupamento



Fonte: Adaptado de Nielsen (2015)

O objetivo desse trabalho é aplicar a deep learning por meio de um modelo de rede convolucional e então avaliar o desempenho desse modelo na análise e categorização de um conjunto de dados de imagem de defeitos de madeira. Como objetivo secundário, busca-se entender qual é o impacto da alteração da quantidade de mapas na primeira camada convolucional e também da dimensão dos campos receptivos locais.

2 MATERIAL E MÉTODOS

Existem várias ferramentas que facilitam o estudo e a criação de modelos de aprendizado de máquina e redes neurais. Muitas dessas ferramentas estão disponíveis em formatos de bibliotecas *python*, que é uma linguagem amplamente usada nessa área. A versão da linguagem *python* utilizada foi a 2.7. Foi utilizada a biblioteca chamada *Keras*, que simplifica bastante a construção, avaliação e validação de modelos de redes neurais e redes convolucionais (CHOLLET, 2015). Para o funcionamento dessa biblioteca, foi necessária a instalação de outras, tais como a *TensorFlow* (ABADI et al., 2016) e *Numpy*

(WALT; COLBERT; VAROQUAUX, 2011). É bastante importante estar atento às dependências para o uso da biblioteca *Keras*.

2.1 Tensorflow

TensorFlow é uma interface que tem como objetivo a elaboração e implementação de algoritmos de aprendizado de máquina. É um sistema bastante flexível e que pode ser usado para expressar vários tipos de algoritmos, o que inclui algoritmos de *deep learning*. Ele é amplamente utilizado para pesquisas e melhora de algoritmos de aprendizado de máquina, o que abrange reconhecimento de voz, visão computacional, robótica, processamento de linguagem natural, extração de informações geográficas e outros. É uma biblioteca em constante desenvolvimento e de código aberto (ABADI et al., 2016). Realiza a computação numérica de mapas de fluxo de dados trabalhando com matrizes multidimensionais dinâmicas que interagem com os nós do mapa, que em geral implementam modelos de operações matemáticas (TENSORFLOW, 2016).

2.2 NumPy

Apesar de a linguagem *python* possuir vários recursos e ser bastante poderosa, ela não é ideal para computação numérica de alta performance. Por esse motivo foi desenvolvido uma estrutura de dados para a computação eficiente de *arrays*, que ficou conhecida como *NumPy*. A biblioteca *NumPy* abrange tanto essa estrutura de *arrays* quanto um conjunto de funções matemáticas. Um *array NumPy* é uma coleção multidimensional de elementos, podendo ter até 32 dimensões e conter diferentes tipos e combinações de elementos *Numpy* (WALT; COLBERT; VAROQUAUX, 2011).

Nesse trabalho, a biblioteca *NumPy* permitiu o carregamento dos conjuntos de dados que foram analisados pelo modelo convolucional.

2.3 Keras

Keras é uma biblioteca de alto nível para redes neurais, pode trabalhar utilizando tanto recursos do *TensorFlow* quanto do *Theano* (AL-RFOU et al., 2016), que têm funcionalidades parecidas com o *Tensorflow*. Foi desenvolvido com foco na experimentação rápida, fazendo com que seja relativamente mais rápido atingir alguns resultados a partir do momento da geração da ideia, aumentando a velocidade de

pesquisas. Tem como objetivo permitir a prototipagem de forma fácil e rápida (CHOLLET, 2015).

2.4 Matplotlib

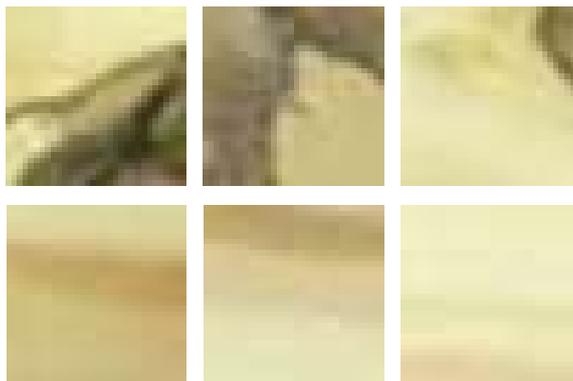
Outra ferramenta utilizada foi a biblioteca *python* chamada *matplotlib*. Essa ferramenta foi utilizada para a geração de gráficos de exibição dos dados obtidos durante os treinamentos e avaliações dos modelos de redes convolucionais. Ela tem como objetivo gerar figuras 2D que ilustrem de forma adequada informações para pesquisas e publicações. Com ela é possível gerar planos, histogramas, espectros de força, gráficos de barra, gráficos de erro e outros (HUNTER, 2007). Os gráficos de dados obtidos por meio da ferramenta *keras* serão exibidos nesse trabalho utilizando-se dessa ferramenta.

2.5 Conjunto de dados

O conjunto original das imagens tinha uma população total de 408 imagens de tábuas de madeira com tamanhos variados (GOMES, 2013). Posteriormente, outro estudo subdividiu as imagens em blocos menores de 128x128, 64x64 e 32x32 *pixels* (ALMEIDA, 2014). O conjunto de dados utilizado é um conjunto de imagens de madeira, que já estão previamente divididas em duas categorias diferentes: madeiras defeituosas e madeiras sem defeito. Ao todo ele possui 440 imagens, sendo que 293 são reservadas para o treinamento do modelo convolucional e 147 para teste do modelo. Essas imagens são RGB, ou seja, possuem 3 canais de cores, algo importante a se saber para a construção do modelo. Elas possuem dimensão de 32x32 *pixels*, sendo uma seleção das imagens do estudo anterior.

Ambos os conjuntos, treinamento e teste, são divididos entre madeiras com defeito e sem defeito. A Figura 2 mostra alguns exemplares de imagens de madeiras defeituosas (linha superior) e imagens sem defeito (linha inferior). No caso do conjunto de 293 imagens de treinamento, 160 imagens são exemplares de madeira limpa, ou seja, sem defeito, e 133 imagens são exemplares de madeira com defeito. Para o conjunto de 147 imagens de teste, 67 são exemplares de imagens de madeira com defeito, e as outras 80 são exemplares de madeiras limpas.

Figura 2. Exemplos de imagens do Conjunto de dados de madeiras



Fonte: Gomes (2013) e Almeida (2014)

2.6 Modelo de Rede Convolutacional

Para a construção do modelo foi necessário analisar as características das imagens trabalhadas. Cada imagem do conjunto de dados possui uma dimensão de 32×32 pixels, sendo que são imagens coloridas RGB. Dessa maneira, o modelo de treinamento foi definido com 3 unidades de entrada de 32×32 neurônios (total de 3072 inputs, sendo 1024 unidades por banda de cor - RGB). Todas as imagens do conjunto de dados foram processadas pela rede convolutacional em épocas, sendo que a cada ciclo em que todas as imagens do conjunto são processadas completa-se uma época de treinamento.

O modelo utilizado foi adaptado de um outro chamado pelo autor de *larger_model* e já obteve como resultado 0.89% de erro no processamento de um conjunto de dados voltado para estudos conhecido como *Mnist* (BROWNLEE, 2016a), basicamente as diferenças entre esse modelo e o utilizado foram a quantidade de mapas de característica e a dimensão dos campos receptivos locais na primeira camada de convolução.

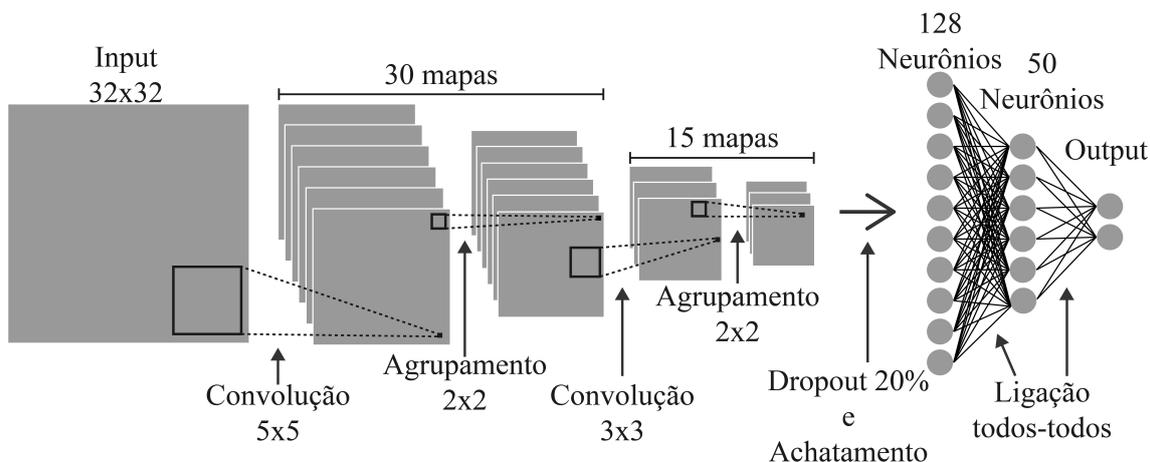
Uma das camadas do modelo realiza o chamado *dropout*, que é uma solução que propõe a não utilização de uma certa porcentagem dos *outputs* da camada anterior na próxima camada do modelo, visando prevenir o fenômeno do *overfitting*. Esse, por sua vez, acontece quando o modelo começa a utilizar muitos parâmetros aprendidos durante o treinamento para a categorização. Isso faz com que os resultados obtidos nas amostras do treinamento sejam muito bons, mas quando se tenta analisar um conjunto de imagens de fora do conjunto de treinamento, um conjunto de teste, por exemplo, o modelo passa a produzir resultados ruins, o que diminui sua praticidade. O *dropout*, portanto, faz com que os resultados obtidos no teste do modelo de rede convolutacional sejam melhores (SRIVASTAVA et al., 2014).

Outras camadas importantes do modelo são as camadas de achatamento e as camadas conectadas na estrutura todos com todos, semelhante a redes neurais artificiais

simples. Essa camada de achatamento transforma a estrutura dos *outputs* bidimensionais que vêm das camadas anteriores em uma estrutura linear. Essa estrutura linear pode ter em sua sequência outra camada com estrutura linear ou a camada de *output* do modelo, onde o resultado final é obtido.

Na topologia, tem-se um modelo sequencial, onde a camada 1 é uma camada de convolução com 30 mapas de característica, campos receptivos locais com dimensão de 5×5 e um *input* tendo o formato de 32×32 *pixels* e 3 canais de cores; a camada 2 é um agrupamento usando um campo de dimensão 2×2 ; a camada 3 é uma convolução com 15 mapas de características e campos receptivos locais de 3×3 ; a camada 4 é um agrupamento com um campo de 2×2 ; a camada 5 estabelece um *dropout* da camada 4 para a 6 de 20%; a camada 6 é uma camada de achatamento; a camada 7 é uma camada de conexão todos com todos de 128 neurônios; a camada 8 é uma camada de conexão todos com todos de 50 neurônios e a camada 9 é a camada de *output* com o resultado do modelo. Esse modelo foi treinado e testado em 50 épocas. A Figura 3 exemplifica o modelo utilizado.

Figura 3. Modelo de rede convolucional utilizado



A ideia era entender como os elementos das camadas convolucionais afetam o resultado final, portanto foi definido pequenas variações na camada 1 do modelo. Com os campos receptivos locais fixos numa dimensão de 5×5 , foram variadas as quantidades dos mapas de característica, nos valores de 20, 30 e 40. Após três execuções do modelo para cada uma das variações nos valores do mapa de característica, foi feito o teste no melhor modelo variando a dimensão dos campos receptivos locais nos valores de 4×4 e 6×6 , também com três execuções para cada um desses modelos, tendo em mente que já havia sido executado com o valor de 5×5 . Dessa maneira, foi possível avaliar o desempenho da

deep learning na categorização de dados de madeira e entender os mecanismos das redes convolucionais.

Os resultados obtidos foram exibidos por meio de tabelas e gráficos gerados pela biblioteca *matplotlib* (BROWNLEE, 2016b).

3 Resultados e discussões

Durante a execução dos modelos propostos foram obtidos resultados bastante satisfatórios. Conforme especificado, inicialmente foram analisados três modelos, o primeiro tendo 40 mapas de característica, o segundo tendo 30 mapas de característica e o terceiro com 20 mapas de característica. Esses modelos tinham como dimensão dos campos receptivos locais 5x5. Conforme mostra a Tabela 1, foram observadas variações consideráveis nos resultados para esses modelos.

Tabela 1. Resultado dos modelos com mapas de característica variados

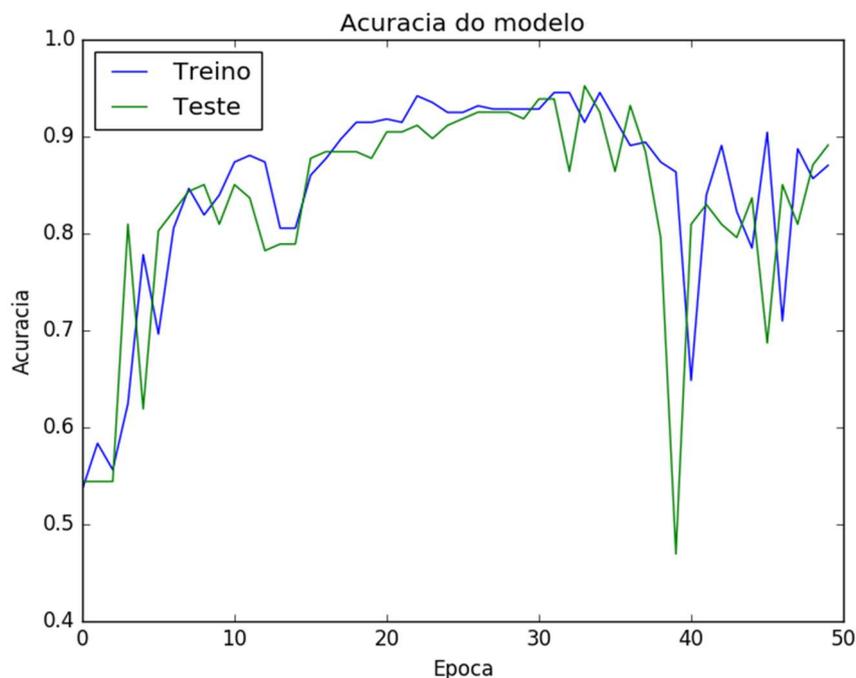
Quantidade de mapas	Dimensão do campo receptivo local	Percentual de Erro			Tempo médio (min.:seg.)
		1ª execução	2ª execução	3ª execução	
40	5x5	10,88%	11,56%	7,48%	1:51.96
30	5x5	14,97%	5,44%	8,84%	1:21.28
20	5x5	6,80%	6,12%	5,44%	1:01.07

Esses resultados mostram que para o conjunto de dados de imagens de madeira que foi utilizado, o aumento da quantidade de mapas de característica na primeira camada gera resultados menos satisfatórios. Isso se dá, provavelmente, devido à pequena quantidade de características que são realmente úteis e relevantes para a categorização correta. O aumento desses mapas de característica força o modelo a analisar características não relevantes para a categorização entre imagens de madeiras com ou sem defeito, fazendo com que o modelo aprenda erroneamente quais características são importantes, gerando uma porcentagem maior de erros.

Pode-se perceber a inconsistência gerada devido à grande variação nos resultados com mais de 20 mapas de característica. A Figura 4 apresenta os gráficos gerados a partir dos resultados obtidos na primeira execução do modelo com 40 mapas de característica, tendo como métrica a acurácia em relação às épocas. É interessante notar que o modelo apresenta resultados ruins pouco depois da trigésima época, o que indica que para

modelos com muitos mapas de característica seria interessante diminuir a quantidade de épocas, para a obtenção de resultados melhores.

Figura 4. Gráfico de acurácia em cada época para o modelo de 40 mapas de características.



Como pode ser observado na Tabela 1, o modelo com 20 mapas de característica foi o que obteve melhores resultado nas três execuções de amostra. Portanto, ele foi o escolhido para a realização dos novos testes, onde foram variadas as dimensões dos campos receptivos locais para 4x4 e 6x6, considerando o resultado já obtido para o dimensionamento 5x5. Os resultados obtidos podem ser observados na Tabela 2. O resultado para os campos de 5x5 foram repetidos para facilitar a visualização.

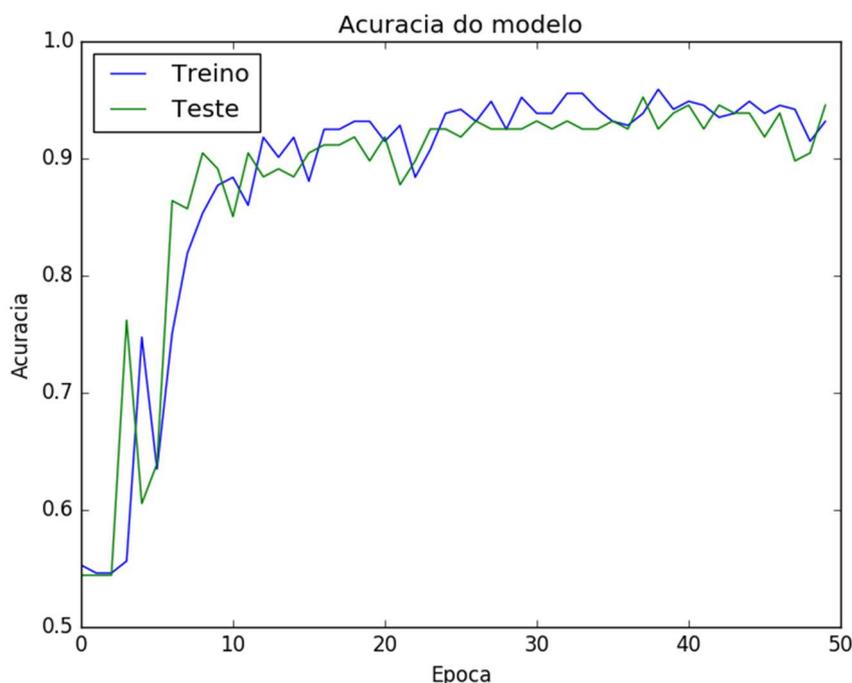
Tabela 2. Resultados do modelo de 20 mapas de característica e campos receptivos variados

Quantidade de mapas	Dimensão do campo receptivo local	Percentual de Erro			Tempo médio (min.:seg.)
		1ª execução	2ª execução	3ª execução	
20	4x4	6,12%	6,12%	5,44%	1:00.28
20	5x5	6,80%	6,12%	5,44%	1:01.07
20	6x6	9,52%	5,44%	18,37%	1:01.97

Como era esperado, o aumento na dimensão dos campos receptivos locais proporcionou resultados piores para o conjunto de imagens aqui usado. Isso não é uma regra para todos os conjuntos, porém, as imagens utilizadas são de 32x32, fazendo com que a região captada nos campos receptivos não aponte adequadamente as características desejadas.

Pelo mesmo motivo, porém, obtiveram-se resultados ainda melhores quando esses campos foram diminuídos para 4x4 nas suas dimensões. Como mostra a Figura 5, os resultados variaram menos, aumentando a confiabilidade do modelo.

Figura 5. Acurácia do modelo de 20 mapas com campos receptivos locais de 4x4



Outro fator observado foi que nos modelos treinados, quanto maior a quantidade de mapas de característica sendo utilizados pelo modelo, mais tempo é utilizado processando as imagens. Esses e os outros resultados já descritos apontam para o modelo onde se tem 20 mapas de característica na primeira camada e campos receptivos locais com dimensão de 4x4.

4 Conclusão

A utilização da *deep learning* para análise e categorização de imagens de tábuas de madeira se provou bastante promissora, tendo apresentado como melhor resultado o valor de 5,44% de erro. Esse sucesso se tornou possível devido às características das redes

convolucionais, que são os campos receptivos locais, o agrupamento, o compartilhamento de pesos entre um mesmo mapa de característica, e, além disso, a organização dos neurônios em formato bidimensional, ou não linear, possibilitando o melhor processamento de imagens 2D.

Para cada conjunto de dados existe uma configuração de topologia que melhor se adapta. A alteração de valores nos mapas de característica e nos campos receptivos locais trazendo resultados diferentes deixaram isso bem claro. Outro ponto é que os modelos apresentam tempos de processamento diferentes, sendo que, dependendo do caso, um modelo pode apresentar um resultado significativamente bom e consistente, porém, utilizar de tanto tempo de processamento que isso torna seu uso inviável.

A *deep learning* possibilita que estudiosos do aprendizado de máquina possam contribuir com pesquisadores de outras áreas de atuação sem necessariamente ter de se especializar naquela área, possibilitando que equipes multidisciplinares sejam mais facilmente formadas. Um pesquisador praticante do aprendizado de máquina, aplicando *deep learning*, não necessariamente precisa aprender quais são as características específicas de um defeito de madeira.

REFERÊNCIAS

- ABADI, M. et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*. *Arxiv*, p.1-19, 16 mar. 2016. Eprint arXiv:1603.04467. Disponível em: <<https://arxiv.org/pdf/1603.04467v2.pdf>>. Acesso em: 19 nov. 2016.
- ALMEIDA, O. C. P. **Classificação de Tábuas de Madeira Usando Processamento de Imagens Digitais e Aprendizado de Máquina**. 105f. Tese (Doutorado), Faculdade de Ciências Agrônomicas de Botucatu – FCA-UNESP, 2014.
- AL-RFOU, R. et al. Theano: *A Python framework for fast computation of mathematical expressions*. *Arxiv*, p.1-19, 9 maio 2016. ArXiv:1605.02688v1. Disponível em: <<https://arxiv.org/pdf/1605.02688.pdf>>. Acesso em: 20 nov. 2016.
- BROWNLEE, J. **Handwritten Digit Recognition using Convolutional Neural Networks in Python with Keras**. 2016a. Disponível em: <<http://machinelearningmastery.com/handwritten-digit-recognition-using-convolutional-neural-networks-python-keras/>>. Acesso em: 24 nov. 2016.
- BROWNLEE, J. **Display Deep Learning Model Training History in Keras**. 2016b. Disponível em: <<http://machinelearningmastery.com/display-deep-learning-model-training-history-in-keras/>>. Acesso em: 24 nov. 2016.
- CHOLLET, F. **Keras: Deep Learning library for Theano and TensorFlow**. 2015. Disponível em: <<https://github.com/fchollet/keras>>. Acesso em: 20 nov. 2016.
- DENG, L.; YU, D. *Introduction: Definitions and Background*. In: DENG, L.; YU, D. **Deep Learning Methods and Applications**. Boston: Now Publishers, 2014. Cap. 1. p. 199-200. (7).
- GOMES, R. C. **Desenvolvimento de uma Base de Dados de Imagens Digitais De Madeira Serrada de Coníferas**. 93f. Dissertação (Mestrado), Faculdade de Ciências Agrônomicas de Botucatu – FCA-UNESP, 2013.
- HASSOUN, M. H. *Threshold Gates*. In: HASSOUN, M. H. **Fundamentals of Artificial Neural Networks**. London: The Mit Press, 1995. Cap. 1, p.1.
- HUNTER, J. D. Matplotlib: *A 2D Graphics Environment*. **Computing In Science & Engineering**, [s.l.], v. 9, n. 3, p.90-95, 18 jun. 2007. *Institute of Electrical and Electronics Engineers* (IEEE). DOI:10.1109/MCSE.2007.55. Disponível em: <<http://dx.doi.org/10.1109/MCSE.2007.55>>. Acesso em: 20 nov. 2016.
- NIELSEN, M. A. *Using neural nets to recognize handwritten digits*. In: NIELSEN, M. A. **Neural Networks and Deep Learning**.: Determination Press, 2015. Cap. 1. Disponível em: <<http://neuralnetworksanddeeplearning.com/chap1.html>>. Acesso em: 20 out. 2016.
- SHALEV-SHWARTZ, S.; BEN-DAVID, S. *Introduction*. In: SHALEV-SHWARTZ, S.; BEN-DAVID, S. **Understanding Machine Learning: From Theory to Algorithms**. New York: Cambridge University Press, 2014. Cap. 1. p. 1-6.

SRIVASTAVA, N. et al. *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*. **Journal Of Machine Learning Research**. Toronto, p. 1929-1958. 15 jun. 2014. Disponível em: <<http://jmlr.org/papers/v15/srivastava14a.html>>. Acesso em: 24 nov. 2016.

TENSORFLOW. *TensorFlow is an Open Source Software Library for Machine Intelligence*. Disponível em: <<https://www.tensorflow.org/>>. Acesso em: 2 nov. 2016.

WALT, S.; COLBERT, S. C.; VAROQUAUX, G. *The NumPy Array: A Structure for Efficient Numerical Computation*. **Computing In Science & Engineering**, v. 13, n. 2, p.22-30, 14 mar. 2011. DOI:10.1109/MCSE.2007.55. Disponível em: <<http://dx.doi.org/10.1109/MCSE.2011.37>>. Acesso em: 24 nov. 2016.

YEGNANARAYANA, B. *Introduction*. In: YEGNANARAYANA, B. **Artificial Neural Networks**. New Delhi: Prentice Hall Of India Pvt. Ltd., 2009. Cap. 1. p. 1-24.

ANEXO A – DIRETRIZES PARA AUTORES

1. SUBMISSÃO DOS TRABALHOS

Deverá ser encaminhada uma declaração de anuência, com nome completo, endereços institucionais e e-mails e as assinaturas de todos os autores, bem como o nome do autor indicado para correspondência, a qual será anexada em “documentos suplementares” no portal da Revista Tekhne e Logos.

O trabalho deve ser acompanhado, se for o caso, de uma declaração de conflito de interesses na qual conste o tipo de conflito.

Todas as instituições patrocinadoras da pesquisa devem ser mencionadas no trabalho.

Toda pesquisa envolvendo seres humanos ou animais deve ter aprovação prévia do Comitê de Ética da instituição de origem. Nesses casos, o número do protocolo no Comitê de Ética deve ser mencionado no trabalho.

As normas da Revista Tekhne e Logos podem sofrer alterações, portanto não deixe de consultá-las antes de fazer a submissão de um artigo. Elas são válidas para todos os trabalhos submetidos neste periódico.

Lembre-se que SE as normas da revista não forem seguidas rigorosamente, seu trabalho não irá tramitar.

2. FORMA E PREPARAÇÃO DOS MANUSCRITOS

Na primeira versão do artigo submetido, os nomes dos autores e a nota de rodapé deverão ser omitidos. Somente na versão final o artigo deverá conter o nome de todos os autores com identificação em nota de rodapé

O manuscrito submetido para publicação deverá digitado em processador de texto em formato DOCX, encaminhado por via eletrônica (<http://www.fatecbt.edu.br/seer>) obedecendo as especificações a seguir:

Papel: formato A4.

Espaçamento do texto: em coluna simples, com espaço entre linhas de 1,5.

Margens: 3,0 cm de margens esquerda e direita. Margens superior e inferior com 2,0 cm, orientação retrato.

Fonte: Times New Roman, tamanho 12.

Parágrafos: 1,25 cm.

Número de páginas: até 15 (quinze) páginas, numeradas consecutivamente, incluindo as ilustrações.

Tabelas: devem fazer parte do corpo do artigo e ser apresentadas no módulo tabela do Word. Essas devem ser elaboradas apenas com linhas horizontais de separação no cabeçalho e ao final das mesmas, evitando o uso de palavras em negrito e coloridas, as quais devem ser ajustadas automaticamente à janela. O título deve ficar acima e centralizado. Se o trabalho for redigido em inglês ou espanhol, deve vir também redigido em português. Exemplo de citações no texto: Tabela 1. Exemplos de citações no título: Tabela 1. Investimento econômico-financeiro (sem ponto no final após o texto). O título deve ficar acima e centralizado, redigido na fonte Times New Roman, tamanho 12. Em tabelas que apresentam a comparação de médias, segundo análise estatística, deverá haver um espaço entre o valor numérico (média) e a letra. As unidades deverão estar entre parêntesis.

Gráficos, Figuras e Fotografias: devem ser apresentados em preto e branco ou em cores (se necessário), nítidos e com contraste, inseridos no texto após a citação dos mesmos, com resolução de 300 dpi. Se o trabalho for redigido em inglês ou espanhol, deve vir também redigido em português. Exemplo de citações no texto: Figura 1. Exemplos de citações no título: Figura 1. Investimento econômico-financeiro (sem ponto no final após o texto). O título deve ficar acima e centralizado, redigido na fonte Times New Roman, tamanho 12(doze).

Fórmulas: deverão ser feitas em processador que possibilite a formatação para o programa Microsoft Word, sem perda de suas formas originais e devem ser alinhadas à esquerda e numeradas sequencialmente à direita

Nomes científicos: devem ser escritos por extenso e em itálico.

3. ESTRUTURA E ORGANIZAÇÃO

3.1 ARTIGO ORIGINAL

O artigo deve ser apresentado na seguinte sequência:

Título: no idioma português com no máximo, 15 (quinze) palavras em letras maiúsculas e em negrito

Autores: até 5 (cinco), por extenso, posicionados logo abaixo do título em inglês ou em português (a depender do idioma do trabalho), com chamada para nota de rodapé da primeira página, com as seguintes informações: formação, titulação e instituição a que o autor está filiado, seguido do endereço, CEP, cidade, estado e endereço de e-mail, sem nenhuma sigla.

Resumo: apresentando em folha à parte, deve condensar, em um único parágrafo, o conteúdo, expondo objetivos, materiais e métodos, os principais resultados e conclusões em não mais do que 250 palavras. A palavra RESUMO deve ser redigida em letras maiúsculas e centralizada.

Palavras-chave: no mínimo de 3 (três) e no máximo de 5 (cinco) termos. Não devem repetir os termos que se acham no título, podem ser constituídas de expressões curtas e não só de palavras e devem ser separadas por ponto em ordem alfabética.

Título: no idioma inglês com, no máximo, 15 (quinze) palavras em letras maiúsculas e em negrito.

Abstract: além de seguir as recomendações do resumo, não ultrapassando 250 palavras, deve ser uma tradução próxima do resumo. A palavra ABSTRACT deve ser redigida em letras maiúsculas e centralizada.

Key words: representam a tradução das palavras-chave para a língua inglesa.

Introdução: Deve ocupar, preferencialmente, no máximo duas páginas, apresentando o problema científico a ser solucionado e sua importância (justificativa para a realização do trabalho), e estabelecer sua relação com resultados de trabalhos publicados sobre o assunto a ser pesquisado. O último parágrafo deve expressar o objetivo, de forma coerente com o constante no Resumo. Esta seção não pode ser dividida em subtítulos.

Material e Métodos: Esta seção pode ser dividida em subtítulos, indicados em negrito. Deve ser redigida com detalhes para que o trabalho possa ser repetido por outros pesquisadores, evidenciando e referenciando a metodologia empregada para a realização da pesquisa e da informação sobre os métodos estatísticos e as transformações de dados.

Resultados e Discussão: Podem ser divididas em subseções, com subtítulos concisos e descritivos. O texto dos Resultados e discussões devem ser discutidos e interpretados à luz da literatura, não apresentando os mesmos resultados das tabelas e figuras.

Conclusões: não devem ser vastas e discursivas, sendo necessário apresentá-las com coerência aos objetivos propostos. Deve ser capaz de evidenciar a solução de seu problema por meio dos resultados obtidos.

Agradecimentos: facultativo.

4. CITAÇÕES NO TEXTO

As citações de autores no texto são conforme os seguintes exemplos:

a) Joaquim (2005) ou (JOAQUIM, 2005)

b) Joaquim e Silva (2010) ou (JOAQUIM; SILVA, 2010)

c) Havendo mais de três autores, é citado apenas o sobrenome do primeiro, seguido de et al. (não itálico): Rossi et al. (2008) ou (ROSSI et al., 2008).

5. REFERÊNCIAS

No artigo deve existir no mínimo dez (10) referências

Devem seguir a NBR 6022, 6021, 6023, 10520, 6028, 6024 da ABNT. Recomenda-se que 70% das referências tenham sido publicadas nos últimos 5 anos e também que 50% sejam de periódicos científicos, apresentadas da seguinte maneira:

a) Artigo de periódico: SIMÕES, D.; SILVA, R. B. G.; SILVA, M. R. Composição do substrato sobre o desenvolvimento, qualidade e custo de produção de mudas de *Eucalyptus grandis* Hill ex Maiden × *Eucalyptus urophylla* S. T. Blake. **Ciência Florestal**, Santa Maria, v. 22, n. 1, p. 91-100, jan/mar. 2012. Disponível em: <<http://dx.doi.org/10.5902/198050985082>>. Acesso: 21 jan. 2014.

b) Livro: MACHADO, C. C.; LOPES, E. S.; BIRRO, M. H. B. **Elementos básicos do transporte florestal rodoviário**. Viçosa: UFV, 2005. 167p.

c) Capítulo de livro: NOGUEIRA, E. Análise de investimentos. In: BATALHA, M. O. (Org.) **Gestão Agroindustrial**. 5. ed. São Paulo, SP. Atlas, 2009. p. 205-266.

d) Dissertação e Tese: MACHADO, R. R. **Avaliação do desempenho logístico do transporte rodoviário de madeira utilizando Rede de Petri**. 75f. Dissertação (Mestrado em Ciência Florestal) apresentada a Universidade Federal de Viçosa/ MG. 2006. Disponível em: <http://www.tede.ufv.br/tedesimplificado/tde_arquivos/4/TDE-2006-11-06T144815Z-43/Publico/texto%20completo.pdf>. Acesso em: 21 ago. 2013.

e) Trabalhos de congressos: SILVA, R. M.; BELDERRAIN, M. C. N. Considerações sobre diagrama tornado em análise de sensibilidade. In: ENCONTRO LATINO AMERICANO DE INICIAÇÃO CIENTÍFICA, 8., 2004, São José dos Campos. **Anais...** São José dos Campos, SP: UNIVAP, 2004. p. 8-11.

f) Trabalhos de conclusão de curso ou monografias: não aceitos.

3.2 ARTIGOS DE REVISÃO

Os artigos de revisão bibliográfica deverão conter: Título (português e inglês), resumo com palavras-chave e abstract com keywords. Introdução; Desenvolvimento do assunto com discussão que deverão ser apresentados em tópicos; Considerações finais e Referências. Deverão conter no máximo 15 páginas.

Botucatu, ____ de _____ de 2017.

Matheus Henrique Baldi de Almeida

De Acordo:

Prof. Me. Roger Cristhian Gomes
Orientador

Prof. Dr. Osvaldo César Pinheiro de Almeida
Co-Orientador

Prof. Esp. Rogério Ferreira Sgoti
Coordenador do Curso de Análise e Desenvolvimento de Sistemas