

## Distributed Artificial Intelligence for Recognition and Position Validation of Printed Circuit Board

JEAN PHELIPE DE OLIVEIRA LIMA<sup>1</sup>  
JOÃO VICTOR MELO DE OLIVEIRA<sup>2</sup>  
RAIMUNDO CORREA DE OLIVEIRA<sup>3</sup>  
Amazonas State University

### Abstract

*This paper presents an Artificial Intelligence system applied to Industry 4.0, which aims at model recognition and positioning validation of printed circuit board on a production line. For the proposed task, two approaches are presented. The first using a single Convolutional Neural Network and the second using a model distributed in two layers of Classifying Neural Networks, wherein the first layer consists of a Convolutional Neural Network and the second one of a set of Multilayer Perceptron Neural Networks. The results of the conducted experiments reached 100% Accuracy in the board recognition task and 88% Accuracy and F1-Score in the positioning validation task, being commercially viable for manufacturing process implantation.*

**Key words:** Artificial Neural Networks, Distributed Artificial Intelligence, Industry 4.0.

---

<sup>1</sup> Graduando em Engenharia de Computação pela Escola Superior de Tecnologia da Universidade do Estado do Amazonas (EST/UEA). Atualmente é pesquisador pelo Programa de Apoio a Iniciação Científica (PAIC) pela UEA e atua em projetos de Pesquisa e Desenvolvimento. Tem experiência na área de Ciência da Computação, com ênfase em Sistemas Embarcados e Inteligência Artificial, atuando principalmente nos temas: Eficiência Energética e Indústria 4.0.

<sup>2</sup> Graduando em Engenharia de Computação pela Escola Superior de Tecnologia da Universidade do Estado do Amazonas (EST/UEA). Atualmente atua em projetos de Pesquisa e Desenvolvimento. Tem experiência na área de Engenharia de Computação, com ênfase em Sistemas Inteligentes e Sistemas Embarcados.

<sup>3</sup> Possui graduação em Engenharia Elétrica Industrial pelo Instituto de Tecnologia da Amazônia (1996), mestrado em Engenharia Elétrica pela Universidade Federal de Pernambuco (2003) e doutorado em Engenharia Elétrica pela Universidade Federal de Pernambuco (2013). Atualmente é professor adjunto da Universidade do Estado do Amazonas. Tem experiência na área de Ciência da Computação, com ênfase em Teleinformática, atuando principalmente nos seguintes temas: fast algorithms, fft, hartley, redes de computador: medidas de segurança, Sistemas Embarcados e fht.

## **Resumo**

*Este artigo apresenta um sistema de Inteligência Artificial aplicado à Indústria 4.0, que visa o reconhecimento de modelos e validação de posicionamento de placas de circuito impresso em uma linha de produção. Para tarefa proposta, são apresentadas duas abordagens, a primeira utilizando uma única Rede Neural Convolutiva e a segunda utilizando um modelo distribuído em duas camadas de Redes Neurais classificadoras, sendo a primeira camada constituída de uma Rede Neural Convolutiva e a segunda de um conjunto de Redes Neurais do tipo Perceptron de Multicamadas. Os resultados dos experimentos conduzidos atingiram 100% de Acurácia na tarefa de reconhecimento de placa e 88% de Acurácia e F1-Score na tarefa de validação de posicionamento, sendo comercialmente viável para implantação em processo fabril.*

**Palavras-chave:** Redes Neurais Artificiais, Inteligência Artificial Distribuída, Indústria 4.0.

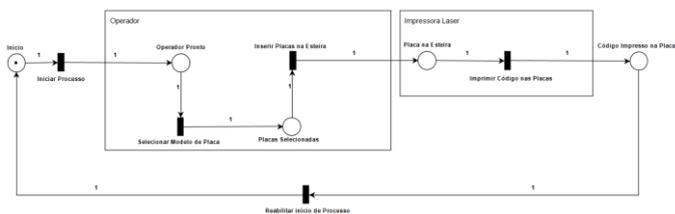
## **1. INTRODUÇÃO**

Indústria 4.0 refere-se ao conjunto de inovações tecnológicas introduzidas pela 4ª Revolução Industrial aplicado ao chão de fábrica [1]. *Smart Factory*, como também é conhecida a Indústria 4.0, é a convergência das seguintes tecnologias: robótica, inteligência artificial (IA), Aprendizado de Máquina, sensoriamento, energias limpas e renováveis, computação em nuvem, Internet of Things (IoT), análise de Big Data, impressão 3D, realidade aumentada, ciber-segurança e integração vertical e horizontal no meio fabril [2]. A fábrica do futuro tem como núcleo a comunicação autônoma entre máquinas, porém ela também foca na colaboração entre a máquina e o ser humano e conecta tudo e todos por meio de sistemas ciber-físicos [3].

Uma das principais tecnologias que compõem a Indústria 4.0 é a IA. Responsável por proporcionar “inteligência” às máquinas, tornando-as hábeis a realizar trabalhos que antes apenas os seres humanos eram capazes de realizar. Uma das aplicações usuais de IA nas fábricas é a automação da inspeção visual, por meio da visão de máquina [4]. Nesse processo são utilizados inputs e outputs digitais

para manipular dispositivos mecânicos em um processo industrial, em que é necessário executar uma ação baseada na análise da imagem feita pelo sistema de visão computacional. Esse método automático de inspeção visual é ideal, por exemplo, para verificação de componentes mal posicionados, erros de impressão de códigos de barras, entre diversas outras aplicações em processos industriais [5]. Assim, visão de máquina desempenha um papel crucial no controle de qualidade dos produtos e o faz com maior eficiência e eficácia que olhos humanos.

Atualmente, em uma das fábricas presentes no Polo Industrial de Manaus (PIM), onde é feita montagem de placas-mãe para computadores, há o seguinte cenário: um operário alimenta uma linha de impressão de códigos de barra com um conjunto de placas de um determinado modelo, sendo que há uma alternância de modelos conforme a necessidade de produção. Esse processo automatizado de impressão de códigos de barras em PCBs é ilustrado por uma Rede de Petri [6] apresentada na Figura 1. A impressão a laser dos códigos de barras deve ser realizada em locais pré-definidos da placa, com isso, o posicionamento incorreto leva a desperdícios no processo de produção. Em média, 6000 placas são produzidas por dia, portanto é um trabalho mecânico e repetitivo passível de erro humano.



**Figura 1: Rede de Petri do processo atual na linha de Impressão de códigos de barras em Placas de Circuito Impresso.**

Esse é um cenário ideal para aplicação de visão de máquina na tentativa de reduzir taxas de erro, em que o método pode ser aplicado antes (verificação da placa) e depois (verificação de códigos de barra) da impressão. Essa proposta de processo é ilustrada pela Rede de Petri apresentada na Figura 2.



Por outro lado, em [11] e [12], os autores combinam técnicas de PDI e RNA para detecção de defeitos em objetos em linhas de produção. Visão computacional utilizando técnicas de Aprendizado de Máquina [13] possui certas vantagens sobre as técnicas de PDI que as tornam mais atrativas, das quais se podem citar: maior precisão em classificação de imagens; requerem expertise e tempo menores dos engenheiros, pois os modelos são treinados e não programados; e os modelos podem ser reutilizados (retreinados) a partir de outras bases de dados [14]. Há também desvantagens: necessidade de grande poder computacional e uma base de dados vasta o suficiente para garantir uma boa capacidade de generalização dos modelos, além de depender da resolução das imagens contidas na base [14].

Em [15] e [16] há uso de redes neurais artificiais para inspeção do posicionamento de componentes eletrônicos em placas de circuito impresso (PCB, do inglês *Printed Circuit Board*). Nesses trabalhos, verifica-se que as técnicas de PDI são aplicadas para realização de um pré-processamento das imagens. Os algoritmos de Aprendizagem de Máquina são alimentados pelos coeficientes numéricos extraídos das imagens, que pode facilitar o processo de aprendizagem e atingir melhores resultados. Em [17], os autores dispensam a etapa de pré-processamento e alimentam uma única CNN diretamente com imagens de sua base de dados para realizar detecção, classificação e verificação de posicionamento de PCBs.

Este trabalho aborda Inteligência Artificial Distribuída para reconhecimento de padrões através de uma base de dados de imagens de diferentes modelos de placas-mãe, utilizando, para isso, técnicas de *Deep Learning* [18], especificamente, Redes Neurais dos tipos Convolutacional e Perceptron de Multicamadas, tendo como objetivo a identificação do modelo e posicionamento correto para a impressão a *laser* de códigos de barras. Com o uso das redes convolucionais, a extração de características, que tradicionalmente é realizada através de técnicas PDI, é realizada de forma inteligente nas camadas convolucionais, enquanto a classificação destas é realizada nas camadas densas. As redes perceptron de multicamadas são utilizadas quando não há a necessidade da extração de características, recebendo como entrada a imagem sem pré-processamento.

### **3. MATERIAIS E MÉTODOS**

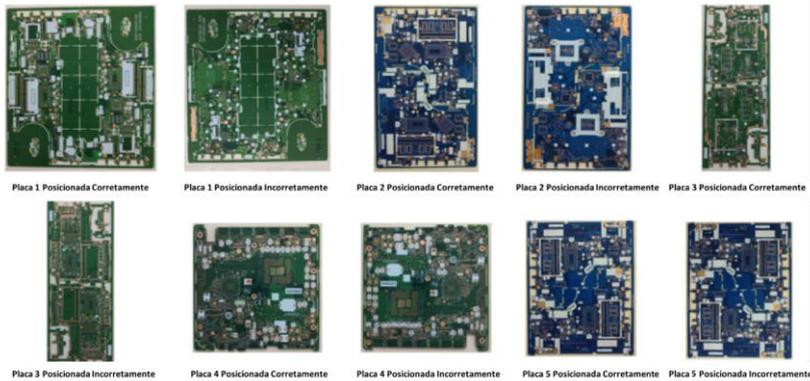
Esta seção descreve os materiais e métodos utilizados para o desenvolvimento e análise dos experimentos conduzidos. As subseções abrangem os seguintes tópicos: Dados Experimentais, que apresenta e analisa a base de dados criada; Descrição da Tarefa de Aprendizado de Máquina, que expõe como o Aprendizado de Máquina será utilizado no contexto deste trabalho em face à base de dados apresentada; em seguida, as duas últimas subseções indicam as abordagens utilizadas, bem como a construção e emprego dos modelos desenvolvidos, além de destacar as diferentes técnicas de pré-processamento de dados utilizadas.

#### **3.1 Dados Experimentais**

Para o experimento em questão, construiu-se uma base de dados contendo exemplos de imagens de 5 diferentes modelos de placas de circuito impresso de uma linha de produção. A captura das imagens foi realizada por meio de uma *webcam* comum, com resolução de até 1280x720 *pixels*. As imagens foram amostradas com resolução de 64x64 *pixels* para economia de poder computacional durante o processamento das mesmas, visto que, em um processo fabril, cada atividade deve consumir o menor tempo possível para aumentar a capacidade produtiva da empresa.

A base de dados, inicialmente, foi dividida em 10 classes: uma classe para cada um dos modelos de placas contendo exemplos das mesmas posicionadas corretamente e uma classe para cada modelo de placa contendo exemplos das mesmas posicionadas incorretamente. Assim, a base de dados foi composta por 5 classes de placas posicionadas corretamente e 5 classes de placas posicionadas incorretamente, totalizando 10 classes.

A quantidade de exemplos amostrados foi bastante restrita, por se tratar de insumos de difícil acesso de uma linha de produção. Dessa forma, coletou-se 15 exemplos de imagens de placas com posicionamento correto para cada modelo de placa e 16 exemplos de imagens de placas com posicionamento incorreto para cada modelo de placa. A Figura 3 apresenta uma amostra de cada uma das 10 classes mencionadas.



**Figura 3: Amostras de Imagens de Placas armazenadas na base de dados, utilizadas para treino e validação de modelos de Aprendizado de Máquina.**

A Figura 3 permite ainda observar que existem características semelhantes entre distintos modelos de placa, que podem ser facilmente confundidas por um humano. Por outro lado, observa-se que cada modelo possui tamanho, forma, cor e outras características bem definidas, indicando assim um possível bom desempenho de técnicas de Aprendizado de Máquina na busca por detecção de padrões nas imagens.

### **3.2 Descrição da Tarefa de Aprendizado de Máquina**

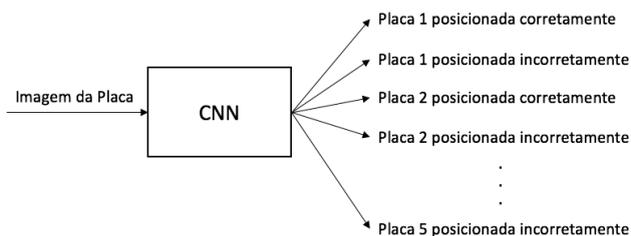
A base de dados será utilizada para realização de um aprendizado de máquina supervisionado com a tarefa de classificação multiclasse de imagens. Tal tarefa será executada em duas etapas: fase de treino e fase de validação.

Na fase de treino, um determinado conjunto de exemplos contido na base de dados será utilizado para geração de um modelo inteligente, capaz de realizar a classificação proposta a partir de novos dados. Para garantir o poder de generalização do modelo, este passará pela fase de validação, que consiste em submeter imagens não contidas no conjunto de treino e avaliar seu desempenho com relação ao atributo alvo desejado para cada imagem.

A fim de manter o conjunto de treino balanceado, definiu-se, aleatoriamente para cada classe, um conjunto de 10 imagens para treino. Destinando, assim, as demais imagens de cada classe para o conjunto de validação.

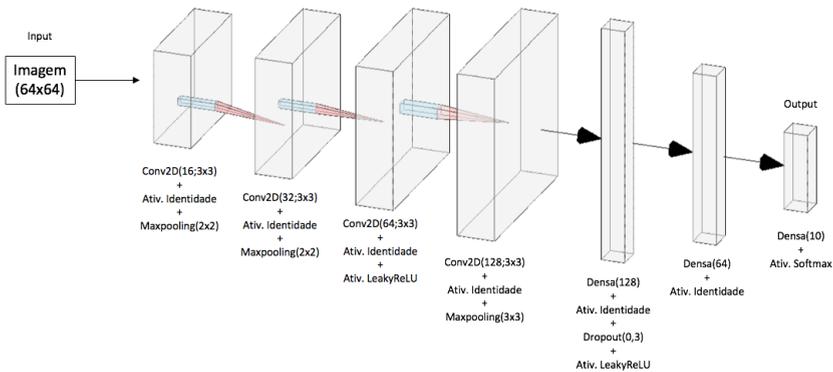
### 3.3 CNN para classificação quanto às 10 classes da base de dados

A primeira abordagem consiste em uma única Rede Neural Convolutacional (CNN, do inglês *Convolutional Neural Network*) para a realização da classificação quanto às 10 classes contidas na base de dados. Isto é, um modelo capaz de reconhecer o tipo de placa e ainda identificar se seu posicionamento está ou não correto. A arquitetura da primeira abordagem é apresentada na Figura 4.



**Figura 4: Arquitetura da primeira abordagem, em que a classificação das 10 classes da tarefa é realizada por uma única CNN.**

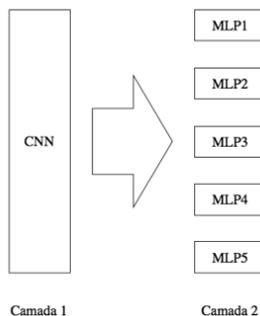
A CNN criada para a tarefa possui a seguinte arquitetura: camada convolutacional com 16 núcleos de tamanho 3x3 e função de ativação identidade; camada *maxpooling* de tamanho 2x2; camada convolutacional com 32 núcleos de tamanho 3x3 e função de ativação identidade; camada *maxpooling* de tamanho 2x2; camada convolutacional com 64 núcleos de tamanho 3x3 e função de ativação identidade; camada de ativação com a função LeakyReLU; camada convolutacional com 128 núcleos de tamanho 3x3 e função de ativação identidade; camada *maxpooling* de tamanho 3x3; camada densa com 128 neurônios e função de ativação identidade; uma camada de regularização de 30% para evitar *overfitting*; camada de ativação com a função LeakyReLU; camada densa com 64 neurônios e função de ativação identidade; e por fim a camada densa de saída com função de ativação *softmax*. A arquitetura da CNN da primeira abordagem é ilustrada pela Figura 5.



**Figura 5: Arquitetura da CNN utilizada na primeira abordagem para as tarefas de reconhecimento e validação de posicionamento de PCBs.**

### 3.4 CNN para reconhecimento e redes MLP binárias para validação de posição

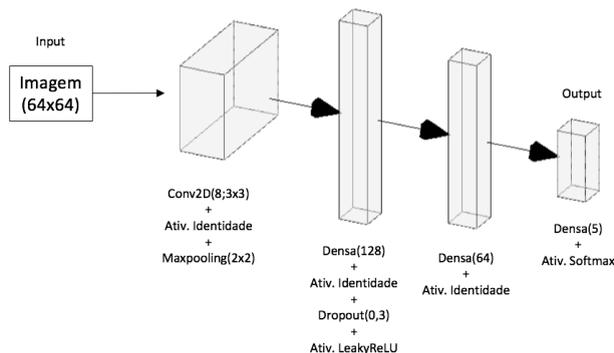
A segunda abordagem consiste em um sistema de Inteligência Artificial Distribuída (IAD) [19], em que o problema foi dividido em dois subproblemas: reconhecimento de modelos de placa; e validação do posicionamento de placa. Utilizou-se uma Rede Neural Convolutacional para a tarefa de reconhecimento de modelo placa, independente do seu posicionamento, e Redes Neurais do tipo MLP para realizar a tarefa de validação do posicionamento da placa. Assim, esta abordagem consiste em duas camadas de redes neurais: a primeira para reconhecimento de placa e a segunda para validação de posicionamento da mesma. A Figura 6 apresenta a arquitetura desta abordagem.



**Figura 6: Arquitetura da segunda abordagem, distribuindo a tarefa para duas camadas de redes classificadoras.**

A Camada 1 representa a CNN multiclasse para reconhecimento de modelo de placa. Para essa tarefa, as 10 classes da base de dados foram interpretadas como apenas 5, ou seja, onde, outrora, havia uma classe para cada placa posicionada corretamente e outra para cada placa posicionada incorretamente, considerou-se, para o treino desta CNN, apenas uma classe para cada modelo de placa contendo suas imagens posicionadas correta e incorretamente.

A CNN criada para a tarefa possui a seguinte arquitetura: uma camada convolucional com 8 núcleos de tamanho 3x3 e função de ativação identidade; uma camada *maxpooling* de tamanho 2x2; uma camada densa com 128 neurônios e função de ativação identidade; uma camada de regularização de 30% para evitar *overfitting*; uma camada de ativação com a função LeakyReLU; uma camada densa com 64 neurônios e função de ativação identidade; e por fim a camada densa de saída com função de ativação *softmax*. A arquitetura da CNN da segunda abordagem é ilustrada pela Figura 7.



**Figura 7: Arquitetura da CNN utilizada na segunda abordagem para reconhecimento de PCBs.**

Após o modelo de placa ter sido identificada corretamente pela Camada 1, a imagem capturada é submetida à Camada 2 a fim de validar seu posicionamento. A Camada 2 é composta por 5 redes MLP binárias. Cada rede foi treinada para um modelo de placa específico, ou seja, cada rede treina para duas classes, sendo uma para a posição correta e outra para posição incorreta de um modelo de placa. Dessa forma, o reconhecimento de placa pela Camada 1 indica qual rede neural MLP será utilizada para validação de posicionamento na Camada 2. Por exemplo, se a Camada 1 classificou uma imagem como

placa do modelo 2, a MLP2, treinada para validar posicionamento de placas deste modelo, recebe a imagem e a classifica binariamente, isto é, informa se o posicionamento está correto ou incorreto. As arquiteturas das redes MLP criadas são apresentadas na Tabela 1. Para cada rede MLP, foi usado como otimizador o lbfgs (*Limited-Memory Broyden-Fletcher-Goldfarb-Shanno*), que é um algoritmo de otimização da família de métodos quasi-Newton que aproxima o algoritmo de Broyden-Fletcher-Goldfarb-Shanno (BFGS) usando uma quantidade limitada de memória de computador, o que é ideal para ajustes de pesos de redes neurais treinadas a partir de bases de dados pequenas [20], como é o caso da tarefa em questão.

**Tabela 1: Arquiteturas das redes MLP utilizadas para validar posicionamento de cada modelo de placa.**

MLP	Número de Camadas	Neurônios por Camada	Função de Ativação	Tamanho de batch	Taxa de Aprend.	Otimizador
1	2	(8,5)	tanh	1	0,05	lbfgs
2	3	(18,12,6)	relu	2	0,02	lbfgs
3	2	(18,8)	relu	2	0,05	lbfgs
4	2	(24,14)	sigmoidal	2	0,05	lbfgs
5	2	(32,16)	sigmoidal	2	0,05	lbfgs

## 4. RESULTADOS E DISCUSSÕES

Nesta seção são descritos os resultados obtidos a partir da metodologia proposta em classificar automaticamente modelos e posicionamento de placas de circuito impresso através da análise de imagens. As subseções abrangem os seguintes tópicos: Desempenho da Primeira Abordagem, que apresenta os resultados obtidos para a tarefa de classificação quanto às dez classes da base de dados; e Desempenho da Segunda Abordagem, que apresenta os resultados obtidos pela divisão da tarefa proposta em duas camadas de RNA.

### 4.1 Desempenho da Primeira Abordagem

Os modelos da primeira abordagem foram treinados e testados para classificar as imagens de placas de circuito impresso quanto ao tipo e ao posicionamento, de acordo com as 10 classes da base de dados. Após os exemplos do conjunto de treino terem sido apresentados à CNN, com intuito de confirmar o aprendizado, realizou-se uma validação:

imagens do conjunto de validação foram submetidos à rede enquanto as respectivas saídas obtidas foram comparadas com as saídas desejadas. O resultado obtido é apresentado na Tabela 2.

**Tabela 2: Resultados da Primeira Abordagem, utilizando uma única CNN.**

Abordagem	Modelo	Acurácia	F1-Score
1	CNN	0,71	0,70

As métricas utilizadas para avaliação do modelo foram Acurácia e F1-Score. Observa-se, na Tabela 2, um resultado razoável do modelo, no entanto longe dos padrões desejáveis industrialmente, em que a taxa de erros deve ser reduzida ao máximo.

#### **4.2 Desempenho da Segunda Abordagem**

A fim de melhorar as métricas obtidas, a segunda abordagem dividiu a tarefa de classificação em duas camadas. A primeira camada consiste em uma rede convolucional que buscou reconhecer apenas o modelo de placa. Os resultados obtidos pela CNN de acordo com as métricas Acurácia e F1-Score são apresentados na Tabela 3. Percebe-se que a tarefa de reconhecimento automático de modelo de placa, isoladamente, é mais fácil que quando associado à validação de posicionamento, o que justifica o aumento considerável de 29% de Acurácia e 30% de F1-Score entre a CNN da primeira e segunda abordagens, em que a última atingiu 100% de Acurácia e F1-Score.

**Tabela 3: Resultados da CNN da Segunda Abordagem, utilizada para reconhecimento de modelo de placa.**

Abordagem	Modelo	Acurácia	F1-Score
2	CNN	1,00	1,00

A segunda camada consiste em uma rede MLP para cada modelo de placa. Cada rede tem a tarefa de classificar uma imagem de placa, já reconhecida pela primeira camada, em relação ao seu posicionamento. As redes foram avaliadas quanto a Acurácia e F1-Score. Os resultados obtidos por cada uma das redes, após a fase de validação, são apresentados na Tabela 4.

**Tabela 4: Resultados obtidos pelas redes MLP da Segunda Abordagem.**

Abordagem	MLP	Acurácia	F1-Score
2	1	0,82	0,82
2	2	0,82	0,82
2	3	0,82	0,82
2	4	1,00	1,00
2	5	0,91	0,91
2	Total	0,88	0,88

Como a Camada 1 não apresentou erros para o conjunto de validação, pode-se afirmar que a Acurácia e F1-Score da segunda abordagem são justamente a Acurácia e F1-Score da Camada 2. Assim, consolida-se a estratégia de distribuição de tarefas entre várias redes neurais, uma vez que houve um aumento de 70% para 88% no F1-Score da primeira para a segunda abordagem.

A Tabela 5 mostra uma comparação com resultados obtidos em [17], que apresenta uma aplicação de CNN para uma tarefa muito próxima à apresentada neste trabalho: os autores utilizam redes neurais convolucionais para tarefa de reconhecimento de modelo e validação de posicionamento de PCBs. Há, no entanto, algumas diferenças de metodologia de [17] em relação a este trabalho, são estas: o número de modelos de placas apresentadas no trabalho relacionado é de 3, enquanto neste é de 5, o que eleva a dificuldade da tarefa apresentada neste trabalho; a câmera utilizada no trabalho relacionado segue um padrão industrial profissional, enquanto neste trabalho foi utilizado uma *webcam* comum; a tarefa de validação de posicionamento apresentada neste trabalho leva em consideração apenas rotações de 180 graus em relação aos eixos x, y e z, enquanto o trabalho relacionado apresenta, além dessas rotações, inclinações constantes em relação ao eixo y (fato que facilita a tarefa de detecção), dessa forma, para fins de comparação de desempenho, utilizou-se o caso em que essa inclinação é mínima pois aproxima ambas as tarefas equiparadas.

**Tabela 5: Comparação de desempenho entre o modelo proposto e modelo apresentado em [17].**

Modelo	Técnica	Reconhecimento Acurácia (%)	Posicionamento Acurácia (%)
[17]	CNN	96,90	89,70
<b>Modelo proposto</b>	<b>CNN + MLP</b>	<b>100,00</b>	<b>88,00</b>

Nota-se, a partir da Tabela 5, que a abordagem apresentada neste trabalho é competitiva à abordagens estado-da-arte para a tarefa proposta, tendo superado, em acurácia, em mais de 3% no reconhecimento de modelo de PCBs e tendo obtido resultados muito próximos na tarefa de validação de posicionamento de PCBs.

## 5. CONSIDERAÇÕES FINAIS

Neste artigo, foram apresentadas duas propostas de reconhecimento de modelos de placas-mãe e seus respectivos posicionamentos espaciais baseadas em algoritmos de *Deep Learning*. O método aplicado dividiu a base de dados disponível em imagens para treinamento e validação, que, por sua vez, foram divididas em 10 classes (referenciando modelos diferentes de placas em condições que resultam em sucesso ou fracasso). Essa base de dados foi utilizada no aprendizado de máquina supervisionado para duas abordagens. A primeira consistiu de uma única CNN para classificação das 10 classes. Na segunda, utilizou-se conjuntamente CNN e redes MLP para, respectivamente, reconhecimento de modelo de placa e validação de posição. O resultado da segunda abordagem mostrou que dividir as tarefas em camadas é a melhor dentre as abordagens apresentadas, atingindo 100% de Acurácia no reconhecimento do modelo e 88% de Acurácia na validação do posicionamento da placa.

Outros algoritmos podem também ser empregados em trabalhos futuros para a realização da tarefa em questão, como técnicas simples baseadas em instâncias, K-Vizinhos mais próximos (KNN, do inglês K-Nearest Neighbors) [21] por exemplo, caso haja garantia de um ambiente controlado, isto é, iluminação constante, distância para câmera constante e posicionamento específico de placa, situação esta completamente possível em ambiente industrial, KNN possivelmente apresentará uma performance competitiva aos resultados obtidos neste trabalho. A utilização de *Deep Learning*, por meio de redes neurais mais profundas que as apresentadas, também deve trazer evolução para a tarefa proposta na tentativa de elevar as métricas de desempenho alcançadas.

## 6. Agradecimentos

Os autores agradecem a empresa Digiboard Eletrônica da Amazônia LTDA pelo apoio financeiro através de projeto de Pesquisa & Desenvolvimento em parceria com a Universidade do Estado do Amazonas.

## REFERÊNCIAS

- [1] K. Schwab, *The Fourth Industrial Revolution*. Crown Business, 1st ed., 2017.
- [2] M. A. K. Bahrin, M. F. Othman, N. H. N. Azli, and M. F. Talib, "Industry 4.0: A review on industrial automation and robotic," in *Jurnal Teknologi*, 2016. DOI: <https://doi.org/10.11113/jt.v78.9285>.
- [3] B. Tjahjono, C. Esplugues, E. Ares, and G. Pelaez, "What does industry 4.0 mean to supply chain?" in *Manufacturing Engineering Society International Conference*, Vigo (Ponvedra), 2017. DOI: <https://doi.org/10.1016/j.promfg.2017.09.191>.
- [4] R. Jain, R. Kasturi, and B. Schunck, *Machine Vision*. McGraw-Hill, 1995.
- [5] R. Szeliski, *Computer Vision: Algorithms and Applications*. Springer, 2010.
- [6] F. Long, P. Zeiler, and B. Bertsche, "Modelling the production systems in industry 4.0 and their availability with high-level petri nets," in *8th IFAC Conference on Manufacturing Modelling, Management and Control MIM*, 2016. DOI: <https://doi.org/10.1016/j.ifacol.2016.07.565>.
- [7] P. Mariño, M. A. Domínguez, and M. Alonso, "Machine-vision based detection for sheet metal industries," in *Electronic Technology Department and Applied Electronics Institute from University of Vigo (Spain)*, 1999. DOI: <https://doi.org/10.1109/IECON.1999.819404>.
- [8] M. Rudek, L. S. Coelho, and O. C. Junior, "Visão computacional aplicada a sistemas produtivos: Fundamentos e estudo de caso," Pontifícia Universidade Católica do Paraná, Laboratório de Automação e Sistemas, 2004.

- [9] J. Jia, "A machine vision application for industrial assembly inspection," in *Second International Conference on Machine Vision*, 2009. DOI: <https://doi.org/10.1109/ICMV.2009.51>.
- [10] S. Sathiyamoorthy, "Industrial application of machine vision," in *International Journal of Research in Engineering and Technology (IJRET)*, 2014. DOI: <https://doi.org/10.15623/ijret.2014.0319120>.
- [11] M. E. Stivanello, "Inspeção industrial através de visão computacional," in *Monografia*, Universidade Regional de Blumenau, Centro de Ciências Exatas e Naturais, 2004.
- [12] L. A. O. Martins, F. L. C. Pádua, and P. E. M. Almeida, "Automatic detection of surface defects on rolled steel using computer vision and artificial neural networks," *IEEE*, 2010. DOI: <https://doi.org/10.1109/IECON.2010.5675519>.
- [13] K. Faceli, A. Lorena, J. Gama, and A. Carvalho, *Inteligência Artificial: Uma Abordagem de Aprendizado de Máquina*. LTC, 1st ed., 2011.
- [14] N. O. Mahony, S. Campbell, A. Carvalho, S. Harapanahalli, G. V. Hernandez, L. Krpalkova, D. Riordan, and J. Walsh, "Deep learning vs. traditional computer vision," in *Advances in Computer Vision, CVC 2019, Advances in Intelligent Systems and Computing*, Springer, Cham, 2019. DOI: [https://doi.org/10.1007/978-3-030-17795-9\\_10](https://doi.org/10.1007/978-3-030-17795-9_10).
- [15] L. K. Cheong, S. A. Suandi, and S. Rahman, "Defects and components recognition in printed circuit boards using convolutional neural network," in *10th International Conference on Robotics, Vision, Signal Processing and Power Applications*, Springer, Singapore, 2019. DOI: [https://doi.org/10.1007/978-981-13-6447-1\\_10](https://doi.org/10.1007/978-981-13-6447-1_10).
- [16] E. H. Yuk, S. H. Park, C. S. Park, and J.-G. Baek, "Feature learning-based printed circuit board inspection via speeded-up robust features and random forest," *Applied Sciences (Switzerland)*, 8(6), [932], 2018. DOI: <https://doi.org/10.3390/app8060932>.
- [17] L. Haochen, Z. Bin, S. Xiaoyong, and Z. Yongting, "Cnn-based model for pose detection of industrial pcb," 2017 10th International Conference on Intelligent Computation Technology

- and Automation (ICICTA), 2017. DOI: <https://doi.org/10.1109/ICICTA.2017.93>.
- [18] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, 2015. DOI: <https://doi.org/10.1038/nature14539>.
- [19] C. G. C. Junior, R. G. Nogueira, and C. D. N. Vinhal, "Inteligência artificial distribuída: Conhecendo para aplicar," 2008. DOI: <http://dx.doi.org/10.18224/est.v35i2.644>.
- [20] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, "A limited memory algorithm for bound constrained optimization," in *SIAM Journal on Scientific Computing*, 1995. DOI: <https://doi.org/10.1137/0916069>.
- [21] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, "Knn model-based approach in classification," in *The Move to Meaningful Internet Systems* (R. Meersman, Z. Tari, and D. Schmidt, eds.), Springer, 2003. DOI: [https://doi.org/10.1007/978-3-540-39964-3\\_62](https://doi.org/10.1007/978-3-540-39964-3_62).