

Atividade Final

Eduardo Satiro da Cruz

PPGMMC

CEFET-MG

Belo Horizonte, Brasil

eduardo.satiro@gmail.com

Abstract—Esse documento mostra a execução de uma rede neural convolucional para prever dígitos de zero a nove. Foi executado trinta vezes e teve uma perda média de 0.09 e acurácia média de 0.99.

Index Terms—CNN, visão computacional, imagens

I. INTRODUÇÃO

Nos últimos anos, houve um aumento na utilização de técnicas de inteligência computacional para resolver problemas de regressão e classificação. As técnicas de aprendizagem foram muito importante para o sucesso de aplicativos de reconhecimento de padrões, como reconhecimento de imagens, voz, escritas.

Esse artigo tem objetivo utilizar a técnica de *Convolutional Neural Network* (CNN) em um conjunto de dados de imagens para realizar a previsão. Para isso foi utilizado um *dataset* através do *Kaggle* [1]. A implementação do algoritmo foi realizada utilizando a linguagem de programação Python versão 3.9, devido à sua popularidade e suporte a bibliotecas de processamento de dados.

A. Conjunto de Dados: Digit Recognizer

O *dataset* é o *Digit Recognizer* [2], esses dados foram fornecido para uma competição do site *Kaggle* [1], o objetivo é identificar corretamente os dígitos de um conjunto de dados de dezenas de milhares de imagens, de acordo com o site. Foi fornecidos dois arquivos, um arquivo para treinamento e um arquivo para teste, contêm imagens em escala de cinza de dígitos desenhados a mão entre zero e nove. Cada imagem tem 28 pixels de altura e largura totalizando 784 pixels. Cada pixel tem um valor de pixel indicando claridade ou escuridão desse pixel, com números mais altos significando mais escuro. Os valores são números inteiros entre 0 e 255. Nas figuras abaixo mostra alguns exemplos das amostras. A Fig. 1 demonstra alguns exemplos dos dados.

II. REFERENCIAL TEÓRICO

LECUN *et al.* (1998) [3] propôs e descreveu a arquitetura de uma *Convolutional Neural Network* (CNN). As CNNs são redes neurais especializadas em processamento de imagens, elas são compostas de camadas de convolução seguidas por camadas de subamostragem. É explicado como filtros de convoluções são aplicados para detectar as características específicas nas imagens e as subamostragens são utilizadas para reduzir a dimensionalidade dos dados. Os autores discutem o

treinamento da CNN usando o algoritmo de retro-propagação do gradiente que ajusta os pesos da rede com base no erro calculado entre as saídas previstas e saídas desejadas. Esse artigo foi fundamental para os avanços no campo da visão computacional.

O SILVA JÚNIOR (2022) [4] fez um trabalho que construiu uma implementação de uma rede neural convolucional pulsada com neurônios integra-e-dispara para classificar imagens e verificar as variações em alguns parâmetros de arquitetura da rede no desempenho. A rede desenvolvida teve um bom desempenho em três conjunto de dados e o autor indica que seja promissora que pode ter seu desempenho melhorado se alterar os parâmetros.

ESPOSTO (2022) [5] fez um trabalho com objetivo de desenvolver uma CNN para prever um subtipo específico de câncer de mama a partir de imagens obtidas por ressonância magnética. Ele concluiu que com a utilização dessa rede teve uma precisão de 65%.

COUTRIN (2023) [6] fez o trabalho com objetivo de investigar a aplicação de diferentes modelos de CNN para a classificação automática da dor neonatal a partir de imagens de face de crianças. Ele implementou modelos das arquiteturas VGG-16, ResNet50, SENet50 e Inception-V3, utilizando o aprendizado por transferência, e da arquitetura N-CNN (*Neonatal Convolutional Neural Network*), que não foi previamente treinado. Foi evidenciada a superioridade dos modelos pré-treinados com imagens de face.

OLIVEIRA (2022) [7] propôs um trabalho para desenvolver e avaliar uma rede neural para segmentar imagens de tomografia, das regiões dos rins e tumor, caso existam. Ele utilizou o *Pyramid Pooling Module* em vez das camadas convolucionais da rede U-Net. Utilizou a métricas de *Iou* (*Intersection over Union*) e obteve 0,91 para região do rim e 0,88 para região com tumor.

VALE *et al.* (2022) [8] aplicou metodologia baseada em CNN e algoritmos de aprendizagem profundo para classificar imagens hiperespectrais para reconhecer padrões. De acordo com os autores, o uso de RNCs e algoritmos de aprendizado profundo na classificação de imagens hiperespectrais podem ser amplamente utilizados no mapeamento da cobertura e uso da terra com a finalidade de monitoramento de ocupações desordenadas da terra, sobre tudo em locais de difícil acesso.

Os autores sugerem que a utilização de redes neurais convolucionais e algoritmos de aprendizado profundo para

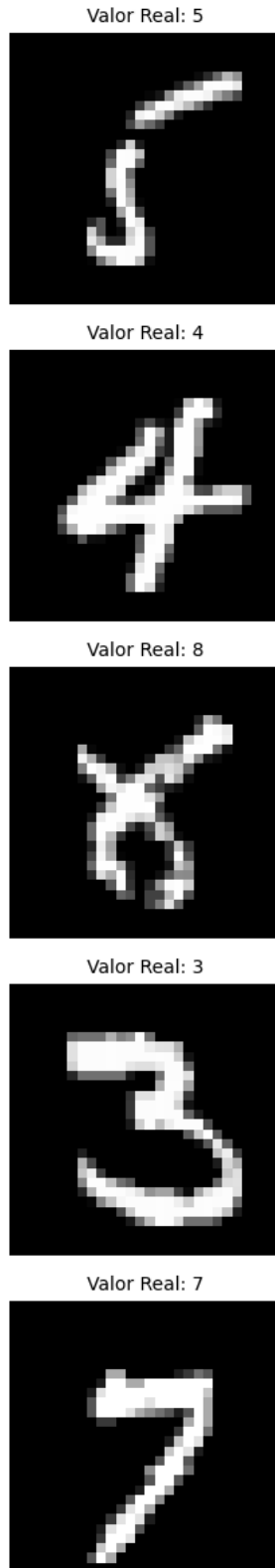


Fig. 1. Exemplos dos dados

classificação de imagens hiperespectrais tem um potencial de aplicação no mapeamento de cobertura e uso da terra, especialmente para monitorar ocupações desordenadas em áreas de difícil acesso.

III. METODOLOGIA

O Kaggle [2] forneceu dois arquivos para a competição, o primeiro foi o 'test.csv' e o 'train.csv' e apenas o segundo tem a coluna informando o número correto então foi utilizado apenas esse documento. Foi utilizado a biblioteca *TensorFlow* [9] para a implementação da rede neural convolucional, a biblioteca *Scikit Learn* [11] para fazer a divisão do *dataset* e fazer a matriz de confusão para análise dos resultados. A biblioteca *Matplotlib* [11] foi utilizada para plotar os gráficos das amostras e resultados. Para a construção da CNN foi construído como:

- 1) Uma camada de convolução 2D com 64 filtros com tamanho de (3, 3) com função de ativação *Rectified Linear Unit* (ReLU), uma função não-linear com a entrada de 28x28 pixels e um canal (imagem em escala de cinza).
- 2) Uma camada de *MaxPooling 2D* com valores (2,2) significando reduzir a resolução da imagem pela metade em cada dimensão. *MaxPooling* é um operação de *pooling* que reduz a dimensionalidade dos dados preservando as características mais importantes.
- 3) Uma camada de convolução 2D com 32 filtros e tamanho de (3, 3) com função de ativação ReLU.
- 4) Uma camada *MaxPooling 2D* com fator de redução (2, 2)
- 5) Uma camada de achatamento ou *flatteing*, essa camada converte a saída da camada anterior para um vetor unidimensional.
- 6) Uma camada *Dropout* com o valor 0.5. É uma técnica para evitar o *overfitting*. O valor 0.5 especifica a taxa de *dropout*, que representa a fração de unidades de saída que serão aleatoriamente definidas como zero durante o treinamento. Nesse caso, 0.5 significa que metade das unidades de saída serão zeradas aleatoriamente durante o treinamento.
- 7) Uma camada *Dense*, com valor de 512, que cria uma camada densa, toda conectada. São 512 neurônios. A função de ativação também foi ReLU.
- 8) Uma última camada *Dense*, com valor 10 que são a quantidade de saídas e função de ativação *softmax* que converte as saídas da camada densa em uma distribuição de probabilidades, onde cada neurônio representa a probabilidade de pertencer a uma determinada classe.

O processo de compilação do modelo utilizou o otimizador ou *optimizer* com o argumento 'adam', a função de perda ou *loss* com o parâmetro '*categorical_crossentropy*' e a métrica ou *metrics* com a *accuracy*.

No treinamento do modelo foi utilizado como parâmetros o conjunto de dados com as variáveis de treinamento, conjunto com as saídas do treinamento, definimos o *batch_size* com 32 que significa o tamanho de lote utilizado durante o treinamento

e o número de épocas ou *epochs* igual a 20 que significa que todo o conjunto de treinamento será percorrido vinte vezes por completo.

O conjunto de dados foi dividido com 80% para treinamento e 20% para teste.

IV. EXPERIMENTOS E ANÁLISE DOS RESULTADOS

Inicialmente foi analisado os dados e não há nenhum dado ausente. Foi verificado a quantidade de imagens para cada dígito tem como resultado:

- 0 - 4132
- 1 - 4684
- 2 - 4177
- 3 - 4351
- 4 - 4072
- 5 - 3795
- 6 - 6137
- 7 - 4401
- 8 - 4063
- 9 - 4188

Na Fig. 2 mostra o gráfico da quantidade de cada dígito.

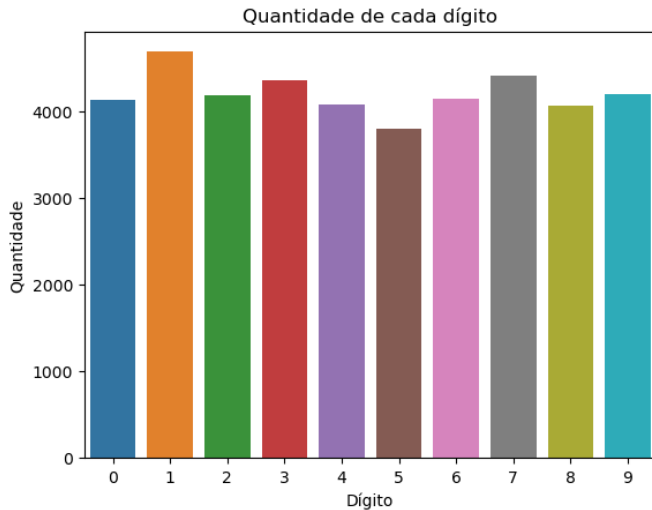


Fig. 2. Quantidade de instância de cada dígito

Como esse modelo é classificado como não determinístico, foi executado trinta vezes. A Fig. 3 consta o *boxplot* da perda médio das trinta execuções.

A Fig. 4 mostra os dados estatísticos da média da perda de execuções.

A Fig. 5 mostra o *boxplot* da média de acurácia das execuções.

A Fig. 6 mostra os dados estatísticos da média da acurácia de execuções.

Na Fig. 7 mostra a matriz de confusão de um dos resultados.

As próximas figuras constam as previsões erradas. Na Fig. 8 o valor real é 5 porém foi previsto o 3.

Na Fig. 9 o valor real também é 5 e o valor previsto era 3.

Na Fig. 10 o valor real é 5 e o valor previsto era 8.

Na Fig. 11 o valor real é 7 e o valor previsto era 4.

Na Fig. 11 o valor real é 9 e o valor previsto era 8.

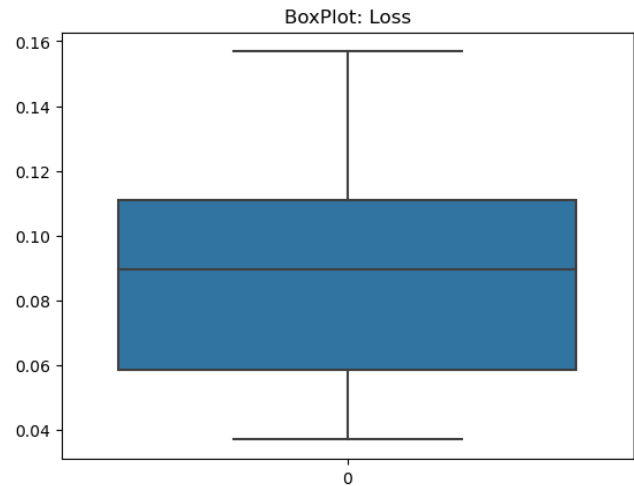


Fig. 3. BoxPlot: Perda

count	30.000000
mean	0.087669
std	0.030951
min	0.037097
25%	0.058634
50%	0.089810
75%	0.110912
max	0.156796

Fig. 4. Análise estatística: perda

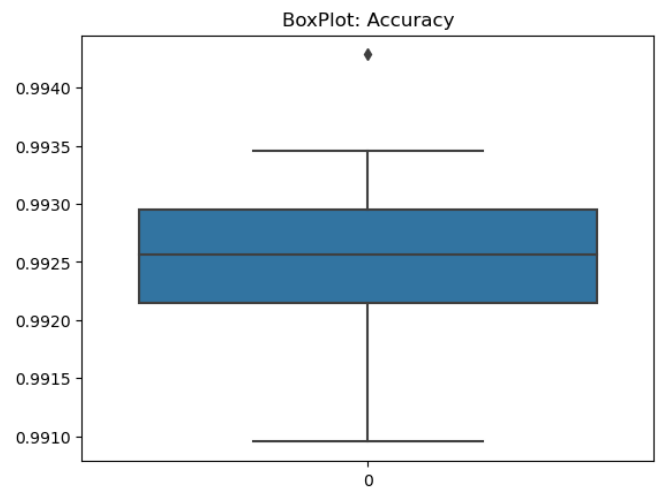


Fig. 5. BoxPlot: Acurácia

count	30.000000
mean	0.992492
std	0.000747
min	0.990952
25%	0.992143
50%	0.992560
75%	0.992946
max	0.994286

Fig. 6. Análise estatística: acurácia

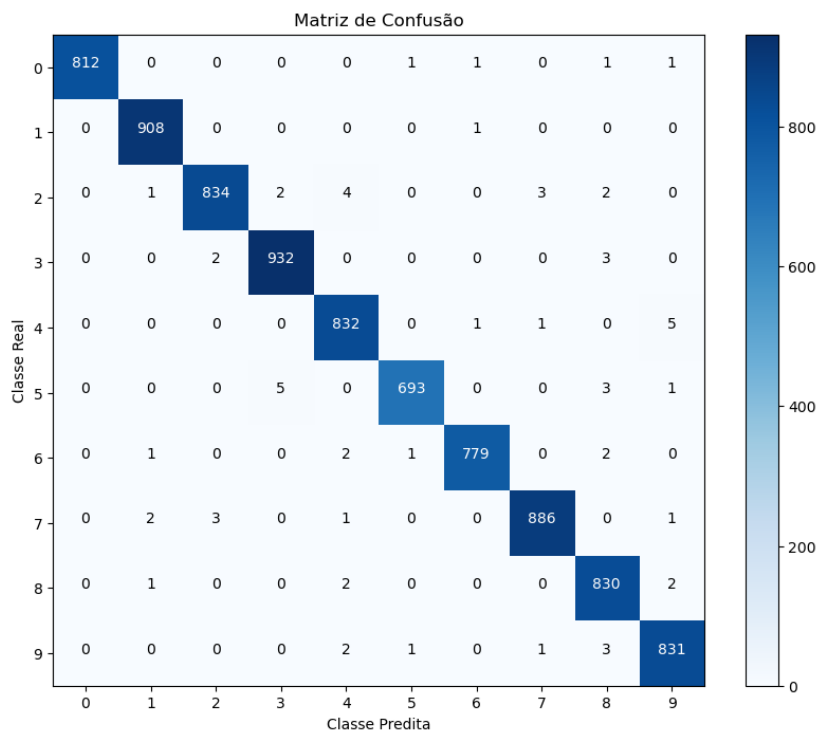


Fig. 7. Matriz de Confusão

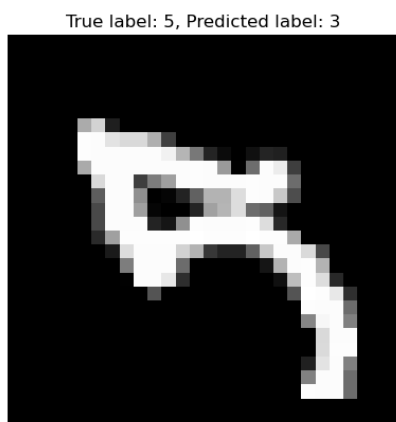


Fig. 8. Previsão 3 e valor esperado 5

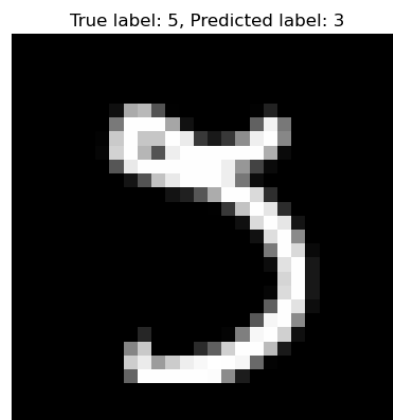


Fig. 9. Previsão 3 e valor esperado 5

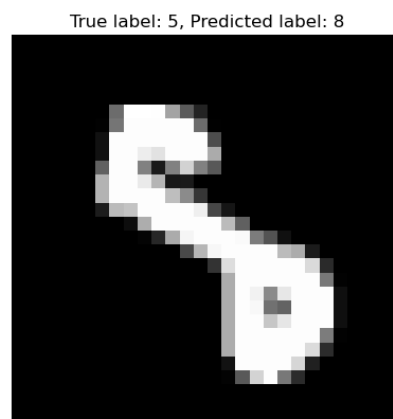


Fig. 10. Previsão 8 e valor esperado 5

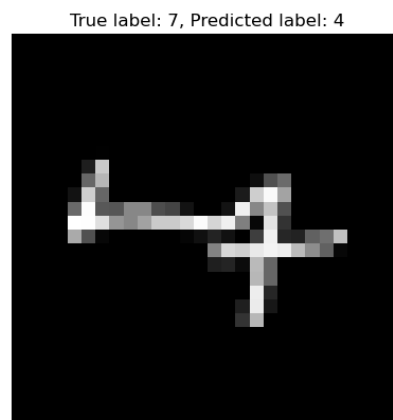


Fig. 11. Previsão 4 e valor esperado 7

True label: 9, Predicted label: 8

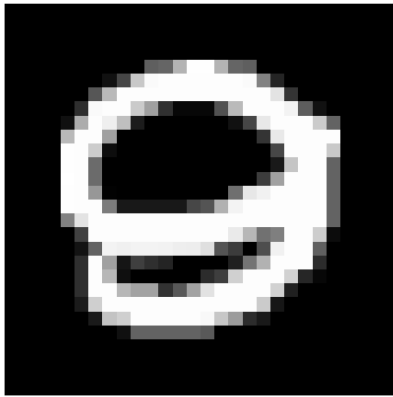


Fig. 12. Previsão 8 e valor esperado 9

V. CONCLUSÕES

Baseado nos resultados desse trabalho e os artigos lidos para o referencial teórico, o uso da técnica CNN e suas variações parece ser promissores para a análises de imagens. Principalmente na área da saúde que utiliza bastante imagens para diagnósticos, a evolução dessa área pode auxiliar bastante os profissionais da saúde para obter resultados significativos.

REFERENCES

- [1] "Kaggle" <https://www.kaggle.com>. Acesso em: 21 mai. 2023
- [2] AstroDave, W. C. (2012). Digit Recognizer. Acesso em: 20 mai. 2023. Disponível em: <https://kaggle.com/competitions/digit-recognizer>.
- [3] Lecun, Y., *et al.* (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324. Disponível em: <https://doi.org/10.1109/5.726791>
- [4] Silva Junior, E. C. (2022). Construção e análise de rede neural convolucional com neurônios pulsados para classificação de imagens. Dissertação (Mestrado em Computação Aplicada) - Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto, University of São Paulo, Ribeirão Preto, 2022. Acesso em: 2023-07-01. Disponível em: <https://doi:10.11606/D.59.2022.tde-22112022-152337>.
- [5] Esposto, F. A. S. (2023). Desenvolvimento de rede neural convolucional utilizando classificadores para predição da agressividade biológica de subtipos de câncer de mama a partir de imagens de ressonância magnética de pacientes de hospital de referência do interior do estado de São Paulo. Universidade Estadual Paulista Júlio de Mesquita Filho, Botucatu, 2023. Acesso em: 1 jul. 2023. Disponível em: <http://hdl.handle.net/11449/243059>
- [6] Couturin, G. A. S. (2023) Redes neurais convolucionais para avaliação de dor neonatal em imagens de face/ uma análise quantitativa e qualitativa. 120 p. Dissertação (Mestrado Engenharia Elétrica) - Centro Universitário FEI, São Bernardo do Campo, 2023. Acesso em: 1 jul. 2023. Disponível em: <https://doi.org/10.31414/EE.2023.D.131609>.
- [7] Oliveira, M. *et al.* (2022). PPM-UNet: Uma Rede Neural Convolucional para a Segmentação de Rins em Imagens de TC. *Anais do XXII Simpósio Brasileiro de Computação Aplicada à Saúde*, Acesso em: 7 jun. 2022. Disponível em: <https://doi.org/10.5753/sbcas.2022.222656>.
- [8] Vale, I. G. *et al.* (2022). Classificação de Imagens hiperespectrais utilizando redes neurais convolucionais para caracterização da ocupação desordenada do solo sobre um recorte do Parque Nacional da Tijuca, RJ. *Revista Brasileira de Geografia Física*, volume 15, 561-571. Disponível em: <https://periodicos.ufpe.br/revistas/rbgfe>.
- [9] Abadi, M. *et al.* (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Acesso em: 20 mai. 2023. Disponível em: <https://www.tensorflow.org/>
- [10] Pedregosa *et al.* (2011) Scikit-learn: Machine Learning in Python, *JMLR* 12, 2825-2830, Acesso em: 7 jun. 2022. Disponível em: <https://scikit-learn.org/>.
- [11] MATPLOTLIB. Matplotlib: Visualization with Python. 2023? Acesso em: 15 jan. 2022. Disponível em: <https://matplotlib.org/>.