Projeto Final - EIA
Estudo sobre CNN e EMNIST

Antônio Vinicius de Moura Rodrigues
Universidade de Brasília (UnB)
antoniovmoura.r@gmail.com
Brasília, Brasil

Bruno Couto Mariño
Universidade de Brasília (UnB)
brunocmarino@gmail.com
Brasília, Brasil

***Abstract*— This project aims to study the implementation of convolutional neural networks (CNN) for image classification from the EMNIST balanced dataset.**
Three articles are presented that were useful for consultation and served as inspiration for the project. Subsequently, there is a description of the applied methodology, which consists of data processing, model configuration, training and presentation of the results obtained through the validation and test sets.

***Keywords*— EMNIST, CNN, Classification**

***Resumo*— Este projeto tem o objetivo de estudar a implementação de redes neurais convolucionais (CNN) para a classificação de imagens do conjunto de dados EMNIST balanced. Ele foi elaborado utilizando o Google Colab e está disponível no GitHub pelo link.**

São apresentados três artigos que foram úteis para consulta e serviram de inspiração para o projeto. Posteriormente, há a descrição da metodologia aplicada, que consiste no tratamento dos dados, na configuração do modelo, treinamento e apresentação dos resultados obtidos através dos conjuntos de validação e teste.

***Keywords*— EMNIST, CNN, Classificação**

I. Introdução

O processo de análise de caracteres escritos a mão consiste em detectar manuscritos e classificá-los em seu determinado rótulo. Para isso, muitas vezes são utilizados algoritmos de inteligência artificial como as redes neurais convolucionais, conhecidas como CNN. Após extrair as imagens e as transformar em digitais, é necessário treinar o algoritmo para que ele aprenda os padrões únicos de cada caractere e classifique-os corretamente.

Um dos conjuntos de dados mais famosos para esse problema é o NIST Special Database 19 [1]. A partir dele, surgiu o MNIST [2], um dataset considerado como padrão para problemas de inteligência artificial e visão computacional. Neste projeto, é apresentado o conjunto de dados EMNIST [3], uma versão extendida do MNIST. Nele, há imagens de letras maiúsculas, minúsculas e números escritos a mão e convertidos para imagens de 28×28 píxeis.

Para estudar mais sobre a base e a metodologia de aplicação do CNN, são apresentados três artigos sobre o assunto. O primeiro busca expandir o dataset EMNIST acrescentando novos caracteres e utilizando CNN para reconhecer as imagens e depois aferir os resultados. O segundo apresenta variações dos conjuntos de dados derivados do NIST e aplica o CNN com o classificador OPIUM. O terceiro, por sua vez, estuda o uso do Hypotheses CNN Pooling para a classificação de mais de um elemento na mesma imagem.

II. Revisão de Literatura

A. Handwritten Character Recognition Using CNN [4]

Esse Artigo tem por objetivo estender o dataset EMNIST adicionando os caracteres da linguagem Tamil. A base utilizada é a EMNIST balanced, que consiste 131,60 imagens de caracteres divididos em 47 classes: números, letras maiúsculas e minúsculas. Vale ressaltar que as letras C, I, J, K, L, M, O, P, S, U, V, W, X, Y e Z, por serem muito parecidas com as minúsculas, são unidas em uma classe apenas.

No pré processamento, as imagens dos caracteres novos foram transformadas em escala de cinza e depois redimensionadas de modo que apresentem a resolução de 28×28. Após isso, é realizado o Feature Extraction para reduzir a dimensão das imagens e extrair a intensidade dos píxeis. Então, é aplicado um redimensionador min-max para ajustar os dados, e por último um normalização com o intuito de facilitar a conversão do modelo.

Por fim, os novos dados são somados ao dataset EMNIST e inseridos em uma rede neural convolucional, a fim de treinar o modelo. Na figura 1 é possível analisar os resultado obtidos no teste.

B. EMNIST: an extension of MNIST to handwritten letters [3]

Esse Artigo tem por objetivo estudar as variações dos datasets EMNIST, aplicando o classificador OPIUM para identificar imagens. Em síntese, os conjuntos estudados

Language	Input data	True Positives	False Positives
Alphabet	26 characters	22	4
Number	10 numbers	9	1
Tamil character	12 characters	9	3

Fig. 1. Retirada do artigo ”Handwritten Character Recognition Using CNN” [4]

podem ser divididos em by_class e by_merge. O primeiro é composto por todas as letras maiúsculas, minúsculas e números; já o segundo é composto por esses grupos, porém com a fusão de algumas letras maiúsculas e minúsculas que são muito parecidas, como por exemplo, ”o” e ”O”.

Em seus resultados, foi possível perceber que o conjunto de dados com 62 classes desbalanceadas obteve o maior nível de erro e o de apenas dígitos obteve o maior nível de acerto. Na figura 2 estão representadas as características dos conjunto de dados e na figura 3 estão representadas as taxas de acerto de cada um.

Name	Classes	No. Training	No. Testing	Validation	Total
By_Class	62	697,932	116,323	No	814,255
By_Merge	47	697,932	116,323	No	814,255
Balanced	47	112,800	18,800	Yes	131,600
Digits	10	240,000	40,000	Yes	280,000
Letters	37	88,800	14,800	Yes	103,600
MNIST	10	60,000	10,000	Yes	70,000

Fig. 2. Retirada do artigo ”EMNIST: an extension of MNIST to handwritten letters ” [3]

	Linear Classifier	OPIUM Classifier
Balanced	50.93%	78.02% ± 0.92%
By Merge	50.51%	72.57% ± 1.18%
By Class	51.80%	69.71% ± 1.47%
Letters	55.78%	85.15% ± 0.12%
EMNIST MNIST	85.11%	96.22% ± 0.14%
Digits	84.70%	95.90% ± 0.40%

Fig. 3. Retirada do artigo ”EMNIST: an extension of MNIST to handwritten letters ” [3]

C. HCP: A Flexible CNN Framework for Multi-label Image Classification [5]

O artigo apresenta uma infraestrutura mais flexível e profunda para o CNN, denominada Hypotheses-CNN-Pooling ou HCP, que é utilizada para identificar mais de um objeto na mesma imagem. Nesse método, um número arbitrário de hipóteses de segmentação de objetos é selecionado e um CNN compartilhado é escolhido para cada um deles. As saídas do CNN são agregadas com o MaxPooling para produzir as predições de mais de um rótulo.

O conjunto de dados explorado pelo artigo é o PASCAL Visual Object Classes Challenge [6] também conhecido como VOC, que é amplamente utilizado como benchmark para classificação multi-rótulos.

A fim de conferir se o modelo é relevante, foi feita uma comparação com outros modelos e métodos, como o modelo I-FT. A partir disso, foi possível perceber que o CNN treinado em larga escala para classificar apenas um rótulo pode ser adaptado para classificar problemas de multi-rótulo. A figura 4 representa uma comparação entre o modelo I-FT e o HCP, feita no relatório apresentado.

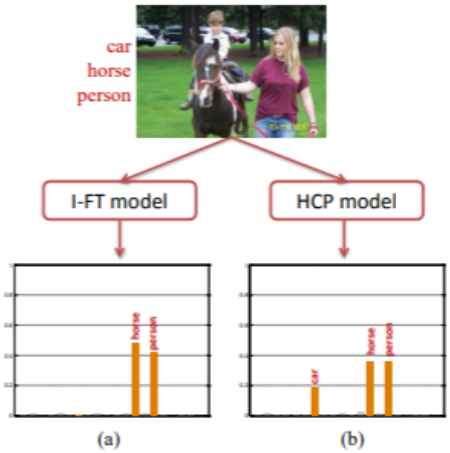

Fig. 4. Retirada do artigo ”HCP: A Flexible CNN Framework for Multi-label Image Classification” [5]

III. Metodologia Aplicada

Este projeto tem o objetivo de estudar a implementação de redes neurais convolucionais para a classificação de imagens do conjunto de dados EMNIST balanced [3]. Para isso, é necessário realizar três etapas: a preparação dos dados, a definição do modelo e a análise dos resultados.

Inicialmente, foi necessário carregar os dados da base utilizada e aplicar uma normalização para reduzir o efeito das diferenças de iluminação, além de fazer com que o modelo CNN convirja com maior velocidade e precisão. No início, as imagens são representadas como um vetor de 784 valores, porém, para facilitar o trabalho e a visualização, elas são remodela-las como matrizes 28×28.

A partir disso, é possível codificar os rótulos que são apresentados como valores de 0 a 47, utilizando o One Hot. Essa codificação permite que a representação dos dados seja mais expressiva, nela, as informações são traduzidas em uma lista na qual as posições dos elementos representam cada rótulo, o número 1 reflete o valor

afirmativo e o 0 negativo.

Após esses processos, os dados foram divididos em 80% para Treino e 20% para Validação.

Na segunda etapa, é definido o modelo CNN. Nele, são realizados os processos de convolução, polling, dropout, flatten e Fully Connected Network. A figura 5, a seguir, representa o sumário do modelo utilizado.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 28, 28, 32)	832
conv2d_1 (Conv2D)	(None, 28, 28, 32)	25632
max_pooling2d (MaxPooling2D)	(None, 14, 14, 32)	0
dropout (Dropout)	(None, 14, 14, 32)	0
conv2d_2 (Conv2D)	(None, 14, 14, 64)	18496
conv2d_3 (Conv2D)	(None, 14, 14, 64)	36928
max_pooling2d_1 (MaxPooling2D)	(None, 7, 7, 64)	0
dropout_1 (Dropout)	(None, 7, 7, 64)	0
flatten (Flatten)	(None, 3136)	0
dense (Dense)	(None, 256)	803072
dropout_2 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 47)	12079

Total params: 897,039
Trainable params: 897,039
Non-trainable params: 0

Fig. 5. Produzida durante a implantação do modelo

Após essa etapa, o modelo foi treinado em 20 épocas, afim de determinar o comportamento da acurácia e da perda, os resultados obtidos podem ser visto na figura 6 e 7.

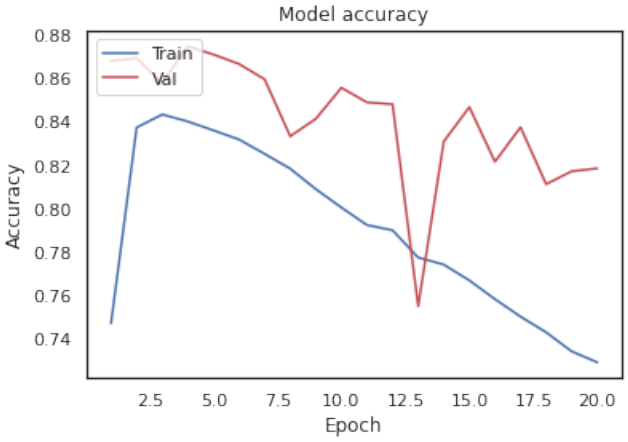

Fig. 6. Acurácia do modelo por época

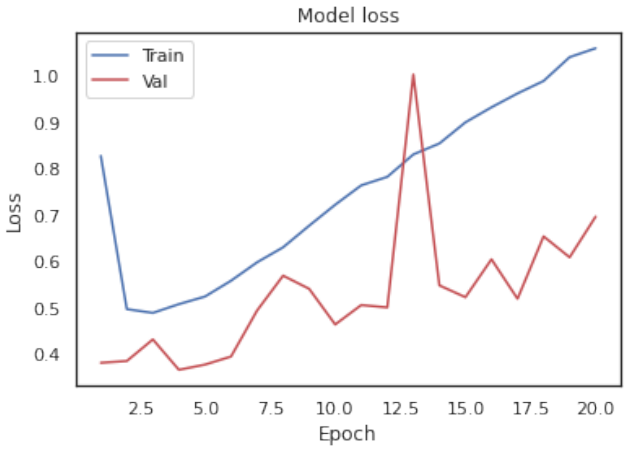

Fig. 7. Perda do modelo por época

A partir desses gráficos de acurácia e perda, é possível perceber que após a terceira época o modelo não é mais tão eficiente e começa a ficar enviesado. Então, o ideal escolhido para seguir com o projeto foi somente 3 épocas.

Na terceira e ultima etapa, são analisados os resultados. Nela tiramos as conclusões sobre a eficácia do modelo e verificamos se há algo a ser melhorado nas etapas anteriores. Essa etapa será melhor elaborada na seção ”V. Resultados”.

IV. Descrição da Base

EMNIST [3] é um conjunto de dados criado com o objetivo de estender os dígitos manuscritos do MNIST [2], nele é possível encontrar imagens com proporção 28×28 que além de incluir números também abrange letras minúsculas e maiúsculas. Existem seis diferentes divisões do EMNIST, porém, nesse artigo o foco é redirecionado ao EMNIST balanceado que contém 131.600 amostras em 47 classes equilibradas, sendo que 10 dessas classes representam os números de 0-9, 26 representam letras maiúsculas A-Z e as demais 11 representam as minúscula que não são parecidas com suas versões maiúsculas.

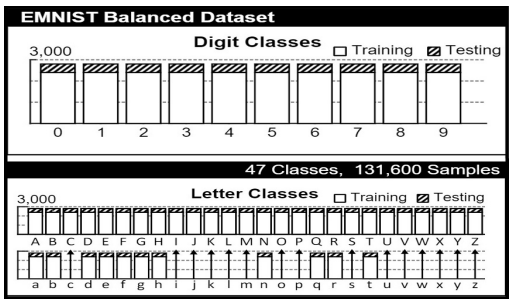

Fig. 8. Adaptada do artigo ”EMNIST: an extension of MNIST to handwritten letters ” [3]

V. Resultados

A fim de analisar os resutlados do modelo CNN criado, foi utilizado o conjunto de 20% de Validação, separado após a preparação dos dados. logo depois de otimizar os hiperparâmetros do modelo, foi possível gerar a matriz de confusão representada na figura 9 e analisar as métricas do modelo representada na figura 10.

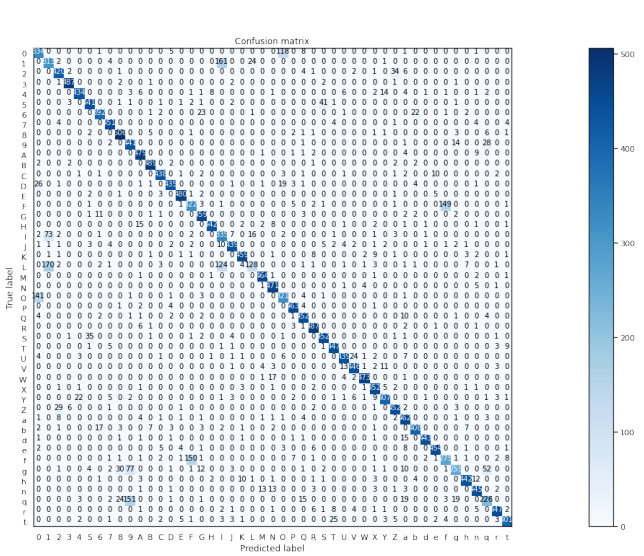

Fig. 9. Matriz de Confusão para o Conjunto de Validação

	precision	recall	f1-score	support
0	0.64	0.71	0.68	468
1	0.56	0.62	0.59	505
2	0.89	0.89	0.89	471
3	0.97	0.98	0.98	497
4	0.93	0.89	0.91	485
5	0.90	0.89	0.89	496
6	0.92	0.88	0.90	445
7	0.94	0.96	0.95	468
8	0.89	0.95	0.92	530
9	0.65	0.91	0.76	487
A	0.92	0.96	0.94	497
B	0.96	0.98	0.97	500
C	0.95	0.95	0.95	463
D	0.95	0.88	0.91	494
E	0.97	0.97	0.97	495
F	0.66	0.66	0.66	487
G	0.89	0.95	0.92	482
H	0.97	0.93	0.95	476
I	0.52	0.75	0.62	446
J	0.94	0.91	0.92	484
K	0.95	0.93	0.94	491
L	0.75	0.28	0.41	451
M	0.95	0.99	0.97	471
N	0.90	0.98	0.94	483
O	0.68	0.68	0.68	473
P	0.95	0.97	0.96	475
Q	0.90	0.94	0.92	480
R	0.93	0.97	0.95	502
S	0.89	0.90	0.90	500
T	0.91	0.96	0.93	468
U	0.93	0.90	0.91	490
V	0.92	0.92	0.92	484
W	0.98	0.95	0.96	498
X	0.91	0.96	0.93	471
Y	0.91	0.87	0.89	466
Z	0.89	0.91	0.90	497
a	0.82	0.94	0.88	491
b	0.92	0.89	0.91	457
d	0.98	0.95	0.97	466
e	0.96	0.94	0.95	485
f	0.63	0.60	0.62	456
g	0.84	0.57	0.68	458
h	0.95	0.92	0.93	481
n	0.91	0.92	0.91	485
q	0.70	0.49	0.58	464
r	0.96	0.93	0.94	480
t	0.93	0.87	0.90	461
accuracy			0.87	22560
macro avg	0.87	0.87	0.87	22560
weighted avg	0.87	0.87	0.87	22560

Fig. 10. Métricas Calculadas para o Conjunto de Validação

Com o modelo treinado e validado, foi possível obter uma acurácia de 87%. Porém, para definir a sua performance de uma forma mais correta, ainda é preciso testá-lo com um conjunto de testes que o sistema nunca teve acesso. Para isso, foi utilizado o emnist-balanced-test [3]. Assim, o resultado da matriz de confusão e das métricas calculadas estão representadas nas imagens 11 e 12.

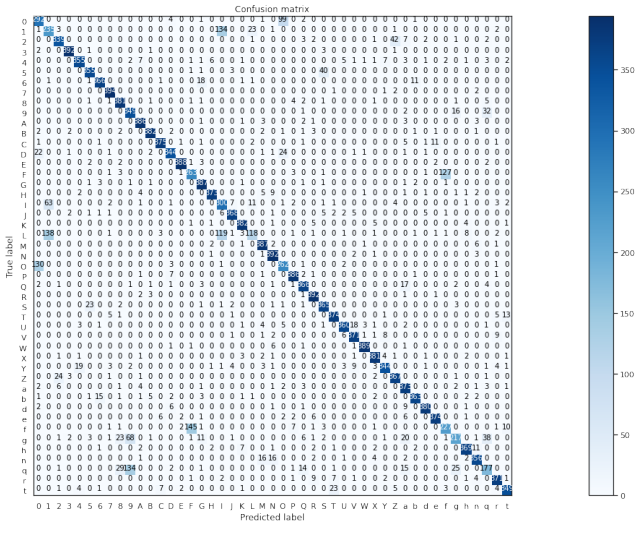

Fig. 11. Matriz de Confusão para o Conjunto de Teste

	precision	recall	f1-score	support
0	0.64	0.73	0.68	400
1	0.54	0.59	0.56	400
2	0.90	0.85	0.87	400
3	0.98	0.98	0.98	400
4	0.92	0.89	0.91	400
5	0.91	0.89	0.90	400
6	0.94	0.92	0.93	400
7	0.96	0.98	0.97	400
8	0.85	0.95	0.90	400
9	0.63	0.87	0.73	400
A	0.94	0.96	0.95	400
B	0.96	0.95	0.96	400
C	0.96	0.94	0.95	400
D	0.92	0.86	0.89	400
E	0.98	0.97	0.97	400
F	0.63	0.66	0.64	400
G	0.89	0.97	0.93	400
H	0.97	0.93	0.95	400
I	0.53	0.75	0.62	400
J	0.95	0.92	0.93	400
K	0.95	0.95	0.95	400
L	0.75	0.29	0.42	400
M	0.91	0.97	0.94	400
N	0.89	0.98	0.94	400
O	0.66	0.66	0.66	400
P	0.95	0.96	0.96	400
Q	0.89	0.92	0.90	400
R	0.92	0.98	0.95	400
S	0.86	0.91	0.89	400
T	0.91	0.94	0.92	400
U	0.95	0.90	0.92	400
V	0.90	0.93	0.92	400
W	0.98	0.97	0.98	400
X	0.94	0.95	0.94	400
Y	0.93	0.86	0.90	400
Z	0.87	0.92	0.89	400
a	0.80	0.93	0.86	400
b	0.95	0.91	0.93	400
d	0.96	0.95	0.96	400
e	0.96	0.94	0.95	400
f	0.63	0.57	0.60	400
g	0.81	0.54	0.65	399
h	0.94	0.92	0.93	400
n	0.91	0.89	0.90	400
q	0.66	0.44	0.53	400
r	0.92	0.93	0.92	400
t	0.92	0.87	0.89	400
accuracy			0.86	18799
macro avg	0.87	0.86	0.86	18799
weighted avg	0.87	0.86	0.86	18799

Fig. 12. Métricas Calculadas para o Conjunto de Teste

Assim, a acurácia do modelo para esse conjunto de teste foi de 86%.

VI. Conclusão

A partir desse projeto, foi possível estudar as diferenças dos conjuntos de dados MNIST e EMNIST, além de entender como a escolha da base pode influenciar nos resultados. Pôde-se Além disso, ficou claro que o modelo possui uma assertividade menor quando precisa diferenciar caracteres parecidos, como a letra ”O” e o número ”0”, a letra ”I” e o número ”1”. Tal dificuldade pode ser relevada, haja vista que boa parte dos humanos também confundem essas letras quando estão isoladas de um contexto, dependendo da forma como estão escritas. O ideal para contornar tal problema seria um algoritmo que entendesse o conjunto em que a frase está escrita. Assim, ele poderia entender que um caractere perto de uma letra é provavelmente outra letra e um caractere perto de um número deve ser um número, por exemplo ”Carro” e ”Carr0” ou até ”120” e ”12o”.

Referências

[1] P. Grother, “Nist special database 19. nist handprinted forms and characters database,” 1970.

[2] L. Deng, “The mnist database of handwritten digit images for machine learning research,” *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[3] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik, “Emnist: Extending mnist to handwritten letters,” pp. 2921–2926, 2017.

[4] S. Anandh Kishan, J. Clinton David, P.Sharon Femi, “Handwritten character recognition using cnn,” September-2018. [Online]. Available: https://www.ijrar.org/papers/IJRAR1903931.pdf

[5] Y. Wei, W. Xia, M. Lin, J. Huang, B. Ni, J. Dong, Y. Zhao, and S. Yan, “Hcp: A flexible cnn framework for multi-label image classification,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 9, pp. 1901–1907, 2016.

[6] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, Jun. 2010.