

Classificação de imagens utilizando o dataset *MNIST*

Alexandre H. Borba
Departamento de Computação
Universidade Estadual de Londrina
Londrina, Brasil
alexandreborba@gmail.com

Abstract—Python *MNIST* recognition using image classification pipeline.

Index Terms—Python, *mnist*, neural network, classification pipeline.

I. INTRODUÇÃO

Seres humanos possuem a habilidade de distinguir e classificar objetos simples, como por exemplo reconhecer a figura de um pássaro e classificar de acordo com sua espécie. Porém quando o objeto a ser classificado se torna mais complexo seres humanos se demonstram limitados a realizar esta tarefa, por exemplo, somos incapazes de determinar com precisão o grau de contaminação de uma planta por uma determinada bactéria, bem como seu grau de amadurecimento e etc ...

Para isto é necessário um processamento computacional desta imagem, geralmente este processamento é feito a partir de um pipeline de classificação de imagens. Este pipeline pode ser subdividido nas seguintes etapas:

- Aquisição de imagens digitais.
- Pré-processamento.
- Extração de características.
- Criação de um modelo.
- Avaliação do modelo.

Nesta atividade foi proposto o reconhecimento de dígitos feitos a mão utilizando o banco de dados *MNIST*.

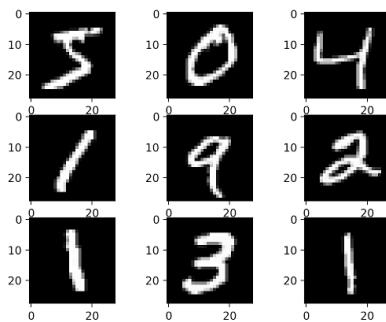


Fig. 1. Exemplo de dígitos do dataset *MNIST*.

A. Aquisição de imagens digitais

O processo de aquisição de imagens foi realizado utilizando o banco de dados *MNIST*.

O *MNIST* é um subconjunto do *NIST Special Database 19*, que foi criado a partir da coleta de dados do Departamento do Censo dos Estados Unidos, contendo um conjunto de dígitos e letras. O *MNIST* contém apenas as imagens referentes a dígitos, sendo tratadas e padronizadas no tamanho de 28x28 pixels.

II. PRÉ PROCESSAMENTO.

O pré processamento é a etapa do pipeline onde a imagem é preparada para os processos seguintes. Na implementação realizada foram destacados os pixels da imagem que compõe o desenho do dígito através da seguinte função `cv2.threshold()` combinada com a flag `THRESH_BINARY_INV`.

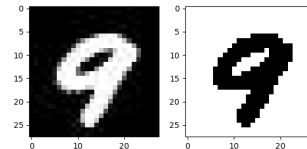


Fig. 2. Resultado obtido ao transformar a imagem original em preto e branco.

Após o formato do numero ser destacado, a imagem também foi recortada, pois alguns dos números presentes no *dataset* estavam deslocados na imagem, podendo interferir no resultado obtido. Desta forma o recorte obtido foi uma imagem 23x23 pixels, pois os valores de saída devem ser padronizados e existiam imagens que ocupavam ao menos 23 pixels na vertical e horizontal.

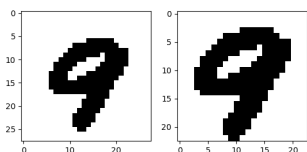


Fig. 3. Imagem recortada

III. EXTRAÇÃO DE CARACTERÍSTICAS.

A etapa de extração de características tem como objetivo extrair dados que tenham a capacidade de representar o objeto de interesse, no nosso caso o dígito. Inicialmente foi desenvolvido um pré processamento aonde após o dígito ser recortado também era subdividido em 4. Após essa subdivisão eram retirados do histograma de cada quadrante que compunha a imagem, os valores estatísticos:

- Média.
- Mediana.
- Moda.
- Assimetria.
- Curtose.

Os valores de media, moda e assimetria foram descartados pois não apresentavam distinção entre as diferentes classes (dígitos). A partir do valores de mediana e curtose foi gerado um modelo, porém não houve convergência. Tendo em vista o fracasso de expressar através de estatísticas as características de uma imagem, foi proposto um novo modelo que aproveitasse o resultado obtido no pré processamento. Desta forma foi aplicada sobre a imagem a transformada de *Fourier* afim de obter o domínio de frequência dos respectivos dígitos.

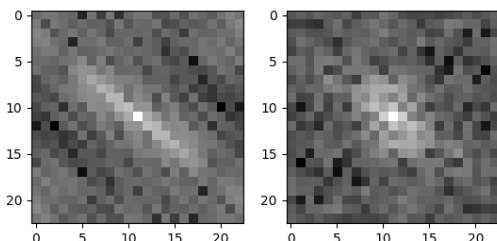


Fig. 4. Comparação entre o fourier do numero 1 e 9

IV. CRIAÇÃO DE UM MODELO

O processo de criação de um modelo consistiu no uso de uma rede neural artificial com ativação sigmoide. Para a entrada desta rede foi utilizado um dataset contendo 1000 imagens, das quais foi extraído o domínio de frequência obtido a partir da transformada de fourier, os pixels resultantes foram representados como um vetor de tamanho 529. A saída desta rede neural é um vetor de 10 posições, onde a posição correspondente a classe do numero é marcada.

0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	1	0	0	0	0

Fig. 5. Exemplo de saída para a classe 5.

Foram feitas várias abordagens em relação ao numero de camadas e quantidade de neuronios por camadas, sendo testado entre 1 até 4 camadas, com quantidade de neuronios variando de 10 a 50, porém não se observou um resultado desejado em nenhuma das tentativas. Entretanto foi constatado que não existiam dados que pudessem ser separados a partir das camadas, pois a entrada do modelo era o domínio de frequência, ou seja, não fazia sentido utilizar mais de uma camada. Assim, foi proposto o modelo com uma camada apenas. Nesta tentativa, apesar da grande variação na acurácia obtida pela rede, é possível observar que utilizando 100.000 iterações a rede acabou convergindo.

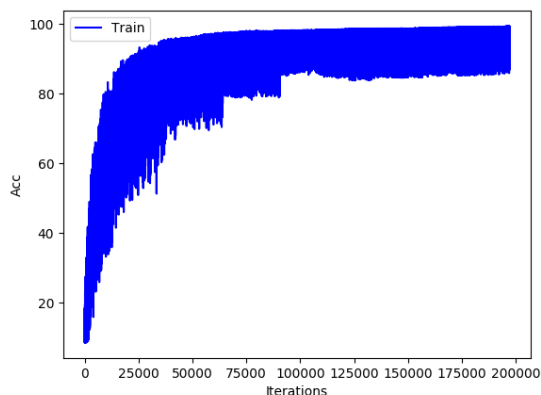


Fig. 6. Treinamento para 100.000 iterações.

V. AVALIAÇÃO DO MODELO.

A avaliação do modelo visa realizar testes utilizando dados aos quais o modelo não foi treinado, para isto foi utilizado um segundo dataset contendo 100 imagens. O modelo obteve uma Acurácia = 41% e a matriz de confusão a seguir:

Predição/Real	0	1	2	3	4	5	6	7	8	9
0	9	1	2	0	0	0	0	0	0	0
1	0	8	0	1	1	0	0	0	0	0
2	4	0	2	2	0	1	1	1	0	0
3	0	0	0	5	0	0	0	0	1	1
4	0	3	1	0	5	0	1	0	0	1
5	0	0	2	1	0	4	0	1	0	0
6	2	0	1	1	2	0	4	1	0	1
7	1	0	2	0	1	0	0	1	0	0
8	2	0	2	1	2	1	1	0	2	0
9	3	1	0	1	1	0	5	1	0	1

Fig. 7. Treinamento para 100.000 iterações.

VI. CONCLUSÃO

Através da utilização do pipeline de classificação de imagens foi possível realizar a classificação de dígitos do banco de dados *MNIST* com uma acurácia de 41%.

REFERENCES

- [1] PYTHON3.7. Fredericksburg, Virginia: Python Software Foundation, 2019. Disponível em:[<https://www.python.org/downloads/>] Acesso em: 26 set. 19.
- [2] Cohen, G., Afshar, S., Tapson, J., van Schaik, A. (2017). EM-NIST: an extension of MNIST to handwritten letters. Retrieved from <http://arxiv.org/abs/1702.05373>