

Impactos da Representação na Classificação de Arquivos de Imagens de Algarismos Arábicos

Thiago Prado de Campos¹

¹Departamento de Informática – Universidade Federal do Paraná (UFPR)
ACF Centro Politécnico – Jd. Das Américas – CEP 81531-980 – Curitiba – PR

contato@thiagotpc.com

Resumo. *Este é um relatório de atividade de laboratório da disciplina de Aprendizagem de Máquina (2020/Período Especial).*

1. A Representação dos Algarismos

A base de dados sob a qual é realizado o laboratório contém dois mil arquivos de imagem (tipo JPEG) em preto e branco de dimensões diferentes que representam os algarismos de 0 a 9, conforme exemplificado a seguir. Acompanha as imagens um arquivo de texto que rotula adequadamente cada arquivo de imagem.











0	1	2	3	4	5	6	7	8	9
									

Figura 1 - Exemplos de imagens na base de dados

2. Atividade Proposta

A atividade proposta é gerar diferentes vetores de características para as imagens, variando as dimensões inicialmente sugeridas para encontrar um conjunto de características que produzem os melhores e os piores resultados de classificação, analisando e comparando suas matrizes de confusão. Adicionalmente, propõe-se tentar obter ainda melhor resultado alterando valores de *neighbor* (k) e métrica de distância.

3. Definindo estratégias para encontrar melhor solução

Em um primeiro momento, para compreender a variação das dimensões presente nos arquivos de dados disponíveis foi criado um script (Quadro 1) que pudesse extrair dos arquivos os valores mínimo, máximo, médio e mediana das larguras e alturas das imagens, bem como um histograma da distribuição dessas dimensões. Também extrai-se a média e mediana da proporção entre largura e altura (*ratio*), com intuito de identificar um valor mais adequado para redimensionamento, considerando um valor próximo ao intervalo de proporção mais usado no conjunto de dados.

Os resultados obtidos por este script são exibidos nas Tabelas 1 e 2 e nas Figuras 2 e 3.

```

# O objetivo deste script é percorrer a pasta de imagens e obter informações para
# gerar um histograma de larguras e alturas usadas na base de imagens, bem como
# obter a média e mediana das larguras e alturas e outras informações.
# Tais informações poderiam trazer algum insight ou servir de base para testes
# com objetivo de melhorar a acurácia do classificador ao determinar o tamanho ideal
# das amostras de treinamento.

import os
from PIL import Image

import statistics

def get_dimensoes_info_from_images():
    count_alturas_usadas = [0]*100
    count_larguras_usadas = [0]*100
    list_alturas = []
    list_larguras = []
    list_proporcao = []

    folder_images = "digits/data"

    for dirpath, _, filenames in os.walk(folder_images):
        for path_image in filenames:
            image = os.path.abspath(os.path.join(dirpath, path_image))
            with Image.open(image) as img:
                width, height = img.size
                count_alturas_usadas[height] = count_alturas_usadas[height] + 1
                count_larguras_usadas[width] = count_larguras_usadas[width] + 1
                list_alturas.append(height)
                list_larguras.append(width)
                list_proporcao.append(width/height)

    print('LARGURA')
    print('min: ', min(list_larguras))
    print('max: ', max(list_larguras))
    print('media: ', statistics.mean(list_larguras))
    print('mediana: ', statistics.median(list_larguras))

    print('ALTURA')
    print('min: ', min(list_alturas))
    print('max: ', max(list_alturas))
    print('media: ', statistics.mean(list_alturas))
    print('mediana: ', statistics.median(list_alturas))

    print('PROPORCAO')
    print('media: ', statistics.mean(list_proporcao))
    print('mediana: ', statistics.median(list_proporcao))

    # histograma
    print('Histograma - Alturas')
    print(*count_alturas_usadas, sep='\n')

    print('Histograma - Larguras')
    print(*count_larguras_usadas, sep='\n')

if __name__ == "__main__":
    get_dimensoes_info_from_images()

```

Quadro 1 - Script para extrair informações das imagens

Tabela 1 - Índices de Largura e Altura extraídos da base de dados

	Largura	Altura
Mínimo	1	19
Máximo	99	81
Média	36,889 (37)	46,621 (47)
Mediana	36	46

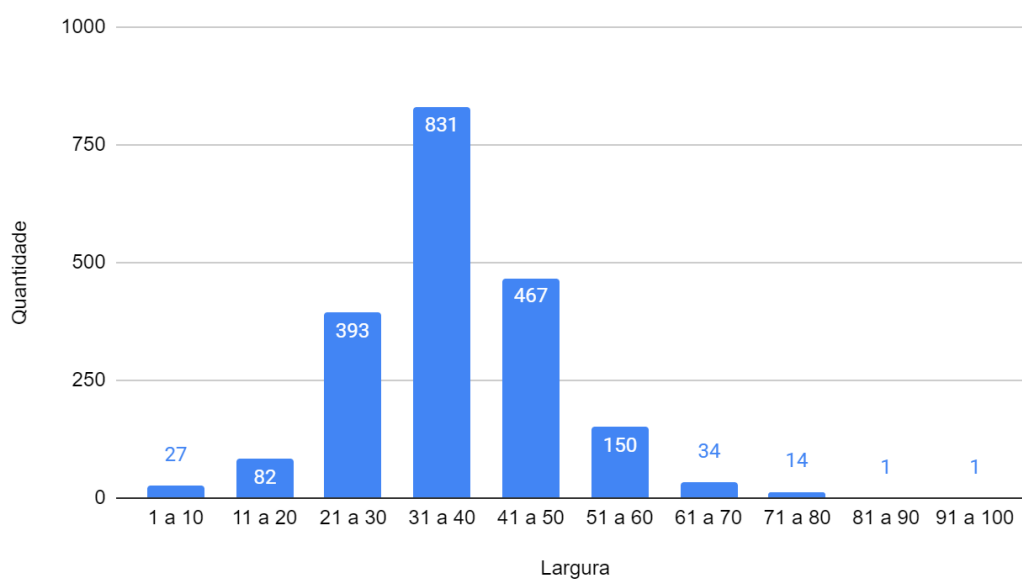


Figura 2 - Histograma de distribuição de largura nos arquivos da base de dados

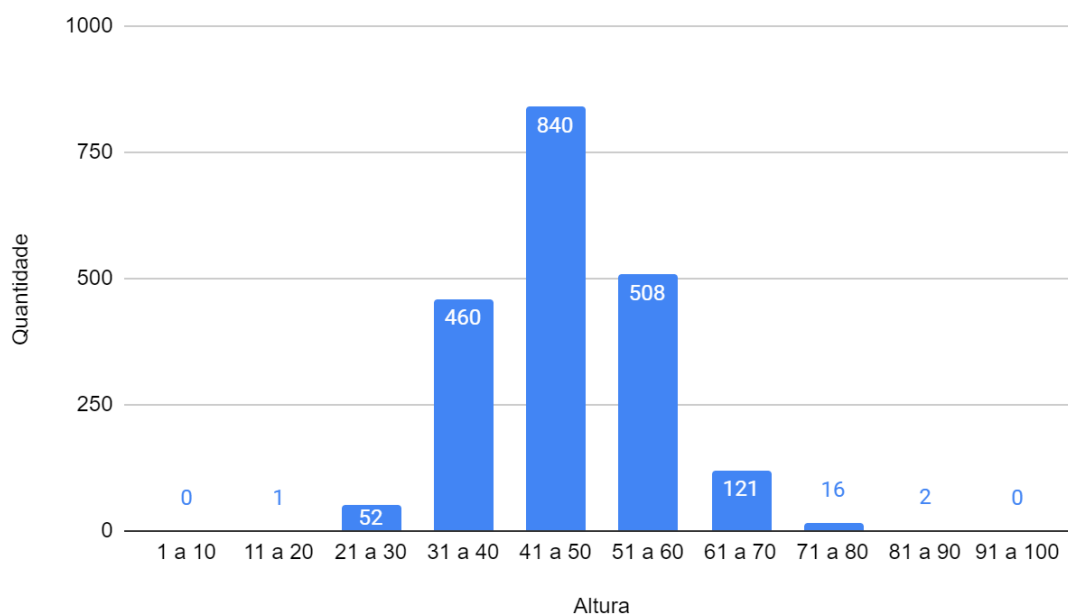


Figura 3 - Histograma de distribuição da altura nos arquivos da base de dados

Tabela 2 - Valores médio e de mediana da proporção das imagens

	Proporção
Média	0,8062
Mediana	0,7777

A partir destes resultados, extraiu-se as seguintes hipóteses de dimensionamento (Tabela 3) da representação. Foi considerado:

- Os valores de média (inteira) e mediana: 37x47 e 36x46;
- O teto das faixas mais frequentes no histograma: 40x50, 50x60, 30x40 e 60x70;
- O teto dos extremos das faixas de pelo menos 50 ocorrências. Para largura: 20 e 60 e para altura: 30 e 70. Portanto, 20x70 e 60x30;
- Dimensão quadrada na menor proporção com uma quantidade razoável de pixels para representação. $10 \times 10 = 100$ pixels;
- Dimensões proporcionais na escala de *ratio* próxima a média (0,8): 12x15, 20x25, 32x40 e 40x50 (que também está contemplada pela faixa de maior frequência).

Em se tratando de imagens, espera-se que quanto mais se reduz o tamanho, mais pode acontecer perda de dados e acarretar erro de classificação. Entretanto, se for possível manter a correta proporção do tamanho, a redução pode não afetar tanto o resultado. E, quanto menor o tamanho, menos custoso será o armazenamento da representação e a execução do treinamento e da classificação.

Tabela 3 - Hipóteses com dimensões e expectativas para representação

#	Descrição	Largura	Altura	Ratio	Expectativa
1	Hipótese Inicial	20	10	2,0000	Ponto de Partida
2	Média	37	47	0,7872	Melhor que inicial
3	Mediana	36	46	0,7826	Melhor que inicial
4	Mais Frequente – 1	40	50	0,8000	Melhor de todas
5	Mais Frequente – 2	50	60	0,8333	Melhor que inicial
6	Mais Frequente – 3	30	40	0,7500	Desconhecida
7	Mais Frequente – 4	60	70	0,8571	Desconhecida
8	Valor Máximo	99	81	1,2222	Pior que o inicial
9	Valores Baixos	10	30	0,3333	Pior que o inicial
10	Extremos Invertidos 1	20	70	0,2857	Pior que o inicial
11	Extremos Invertidos 2	60	30	2,0000	Pior que o inicial
12	Quadrado pequeno	10	10	1,0000	Pior de todos
13	Proporcional 1	12	15	0,8000	Melhor que inicial
14	Proporcional 2	20	25	0,8000	Melhor que inicial

15	Proporcional 3	32	40	0,8000	Melhor que inicial
----	----------------	----	----	--------	--------------------

4. Gerando o vetor de características para cada dimensão de representação

O script *digits.py* fornecido pelo professor foi modificado com objetivo de gerar vetores de representação da base de dados para cada dimensão escolhida. Para isto, criou-se uma lista de hipóteses contendo as larguras e alturas que se desejaria testar. Foi modificado também o script *knn.py* com objetivo de incluir na entrada e na saída informações de *n_neighbors* e métrica de distância utilizada, para automatizar os testes e gerar relatório. As mudanças aplicadas estão disponíveis no projeto no Github¹. Os arquivos de características gerados foram listados na Tabela 4.

Tabela 4 - Arquivos de características de representação e seu tamanho

Arquivo	Tamanho
features 10x10.txt	965KB
features 10x30.txt	3.309KB
features 12x15.txt	1.903KB
features 20x10.txt	2.137KB
features 20x25.txt	5.653KB
features 20x70.txt	16.981KB
features 30x40.txt	14.247KB
features 32x40.txt	15.340KB
features 36x46.txt	20.481KB
features 37x47.txt	21.616KB
features 40x50.txt	25.184KB
features 50x60.txt	38.856KB
features 60x30.txt	22.450KB
features 60x70.txt	55.262KB
features 99x81.txt	107.475KB

5. Primeiros Resultados

Ao executar a primeira rodada de treino e classificação, mantendo *n_neighbors=3* e métrica Euclidiana do script originalmente fornecido, obteve-se os resultados de acurácia mostrados na Tabela 5.

¹ <https://github.com/thiagotpc/ml-laboratorio-01>

Tabela 5 - Resultados para as hipóteses levantadas

#	Descrição	Largura	Altura	Ratio	Expectativa	Acurácia
1	Hipótese Inicial	20	10	2,0000	Ponto de Partida	0,905
2	Média	37	47	0,7872	Melhor que inicial	0,920
3	Mediana	36	46	0,7826	Melhor que inicial	0,919
4	Mais Frequente – 1	40	50	0,8000	Melhor de todas	0,924
5	Mais Frequente – 2	50	60	0,8333	Melhor que inicial	0,919
6	Mais Frequente – 3	30	40	0,7500	Desconhecida	0,909
7	Mais Frequente – 4	60	70	0,8571	Desconhecida	0,924
8	Valor Máximo	99	81	1,2222	Pior que o inicial	0,919
9	Valores Baixos	10	30	0,3333	Pior que o inicial	0,908
10	Extremos Invertidos 1	20	70	0,2857	Pior que o inicial	0,925
11	Extremos Invertidos 2	60	30	2,0000	Pior que o inicial	0,918
12	Quadrado pequeno	10	10	1,0000	Pior de todos	0,894
13	Proporcional 1	12	15	0,8000	Melhor que inicial	0,907
14	Proporcional 2	20	25	0,8000	Melhor que inicial	0,928
15	Proporcional 3	32	40	0,8000	Melhor que inicial	0,913

Confirmou-se como pior resultado (89,4%), a representação quadrada muito pequena (poucos pixels). Todas as outras dimensões apresentaram resultados melhores. Com destaque para a proporção 32x40, que apresentou o melhor resultado (92,8%). Outras dimensões que tiveram bom desempenho foram as de 20x70, 40x50 e 60x70, respectivamente com acurácia de 92,5%, 92,4% e 92,4%. Esperava-se que 40x50 tivesse o melhor resultado pois seria a que possuía mais dimensões de características na proporção próxima a média de *ratio* da base de dados.

Observando a matriz de confusão da hipótese #12 (a pior), identificamos como principal dificuldade a de identificar os algarismos 4 e 9, confundidos, frequentemente com 1 e 7, respectivamente.

```

Accuracy: 0.894
[[ 94  1  0  0  0  1  1  0  0  0]
 [  0 93  0  0  0  1  0  0  0  1]
 [  1  4 100  0  1  0  1  3  1  0]
 [  2  0  3 95  0  0  0  0  2  1]
 [  0 11  1  0 80  0  0  0  0  3]
 [  1  1  0  5  0 89  0  0  1  0]
 [  1  7  0  0  0  0 98  0  0  0]
 [  0  8  1  0  0  1  0 81  0  6]
 [  0  5  0  4  1  1  0  4 69  3]
 [  0  2  0  0  4  0  0 11  0 95]]

```

Figura 4 - Matriz de confusão para Hipótese #12 - 10x10

A matriz de confusão da hipótese #14 (20x25) mostra que foi possível reduzir o erro na classificação do algarismo 4 como 1 (11 para 9 erros) e aumentar o acerto para classificar os algarismos 3, 7, 8 e 9 (95→100, 81→87, 69→75 e 95→101).

Accuracy: 0.928

```

[[ 96  0  0  0  0  1  0  0  0  0]
 [  0 95  0  0  0  0  0  0  0  0]
 [  0  5 101  0  1  0  1  3  0  0]
 [  1  1  0 100  0  1  0  0  0  0]
 [  0  9  2  0 82  0  0  0  0  2]
 [  1  0  0  4  0 91  1  0  0  0]
 [  1  5  0  0  0  0 100  0  0  0]
 [  0  7  0  0  0  0  0 87  0  3]
 [  0  2  1  4  1  2  0  2 75  0]
 [  0  0  0  0  3  0  0  7  1 101]]

```

Figura 5 - Matriz de confusão para Hipótese #14 – 20x25

6. Alterando n-neighbors e métrica de distância

Considerando que quase todas as novas hipóteses foram melhores que a inicial e que havia entre elas uma pequena diferença na acurácia, foi posto à prova novamente todas as hipóteses de representação, mas variando-se desta vez os parâmetros de *n-neighbors* (*k*) e métrica de distanciamento, considerando *k*=1 e *k*=5 e as métricas Euclidiana, Manhattan e Minkowski.

Os resultados foram adicionados aos já existentes. Então, tivemos 12 dimensões, 3 valores para *n-neighbor* e 3 métricas (15x3x3), totalizando 135 possibilidades.

Observando os resultados, vê-se que a variação de *n-neighbor* altera a acurácia. Quando *k*=1 tem-se, em geral, resultados melhores. Por outro lado, a variação da métrica de distância não trouxe diferença nas acurácias calculadas. A seguir listamos os melhores resultados, considerando apenas a métrica euclidiana, para ilustrar o impacto da escolha da dimensão e do número de *n-neighbors*.

Tabela 6 - Resultados quando alterado n-neighbors

#	Descrição	Largura	Altura	N-Neighbor	Métrica	Acurácia
1	Média	37	47	1	Euclidiana	0,930
2	Mais Frequente – 2	50	60	1	Euclidiana	0,930
3	Valor Máximo	99	81	1	Euclidiana	0,928
4	Proporcional – 2	20	25	3	Euclidiana	0,928
5	Proporcional – 3	32	40	1	Euclidiana	0,927
6	Extremos Invertidos 1	20	70	1	Euclidiana	0,926
7	Mais Frequente – 4	60	70	1	Euclidiana	0,926
8	Extremos Invertidos 1	20	70	3	Euclidiana	0,925
9	Extremos Invertidos 2	60	30	1	Euclidiana	0,925

10	Proporcional – 2	20	25	1	Euclidiana	0,925
11	Mais Frequente – 1	40	50	1	Euclidiana	0,924
12	Mais Frequente – 1	40	50	3	Euclidiana	0,924
13	Mais Frequente – 4	60	70	3	Euclidiana	0,924
14	Mediana	36	46	1	Euclidiana	0,923

Ainda com $k=5$ é possível obter melhores resultados com outras dimensões diferentes da inicialmente sugerida, como listada nas hipóteses a seguir.

Tabela 7 - Hipóteses com $k=5$ e melhores resultados que a sugestão de dimensionamento inicial

Descrição	Largura	Altura	N-Neighbor	Métrica	Acurácia
Extremos Invertidos 1	20	70	5	Euclidiana	0,916
Mais Frequente – 2	50	60	5	Euclidiana	0,916
Mais Frequente – 4	60	70	5	Euclidiana	0,916
Valor Máximo	99	81	5	Euclidiana	0,915
Média	37	47	5	Euclidiana	0,914
Extremos Invertidos 2	60	30	5	Euclidiana	0,914
Mediana	36	46	5	Euclidiana	0,912
Mais Frequente – 3	30	40	5	Euclidiana	0,908
Proporcional – 2	20	25	5	Euclidiana	0,908
Mais Frequente – 1	40	50	5	Euclidiana	0,906

Todo o conjunto de resultados está disponível em arquivo CSV no repositório do projeto no Github, bem como as saídas do console de execução dos scripts Python.

7. Considerações Finais

Observa-se, ao final, que para o caso aqui tratado, o dimensionamento da representação pela média da largura e altura do conjunto de dados pode ser uma boa solução, dado o uso de $k=1$. A mesma acurácia pode ser obtida na dimensão 50×60 , também com $k=1$. Outras soluções também são melhores que a hipótese inicial do laboratório (20×10), mesmo quando se tem $k=5$.

Percebe-se que, para esta classificação em particular, não houve ganho quando alterada as métricas de distância (Manhattan ou Minkowski). Pode-se concluir também que reduzir demais a imagem (e o vetor de representação) pode levar a um resultado ruim.

Este laboratório pode ser melhorado criando uma saída tabulada para se evitar a tabulação manual e outras informações podem ser calculados e registradas, como, por exemplo, o F1-Score e o tempo de execução dos scripts ou suas partes (normalização/vetorização/treinamento/classificação). Deste modo, pode-se obter uma nova interpretação para os parâmetros modificados.