

MC940 - Análise de Imagem

Felipe Santana Dias - 215775

19 de julho de 2022

Sumário

Com o objetivo de verificar se um par de impressões digitais correspondem ao mesmo indivíduo, neste relatório foram utilizadas técnicas de processamento de imagem para extrair ROIs (Regions of Interest) combinadas à utilização de SiNN (Redes Neurais Siamesas) para realizar a comparação entre as imagens de impressões digitais,.

O relatório é obedece à seguinte estrutura: Inicialmente é apresentado o problema a ser resolvido e os dados disponíveis para o desenvolvimento da solução. Na seção 2 é introduzida a literatura proposta que embasa os algoritmos criados. Estes são apresentados nas seções seguintes. As seções 5 e 6 englobam a conclusão e as referências, respectivamente.

1. Problema

Um dos desafios do processamento e análise de imagens é a capacidade de extrair informações desses arquivos e interpretá-las. No caso das impressões digitais, diferentes imagens em diferentes posições e com diferentes métodos de extração de um mesmo indivíduo precisam ser identificados corretamente.

Assim, a comparação correta entre pares de impressões digitais - comparando entre fonte e destino - é a tarefa crucial e complexa que nos propomos a explorar nesse projeto.

1.1 Dados utilizados

Os algoritmos utilizados foram desenvolvidos especificamente para a base de dados fornecida na pasta *images_02*. A classificação e rotulação das imagens utilizadas para treinamento e validação foram feitas baseando-se nas nomenclaturas existentes. As imagens possuem o seguinte padrão de nomeação:

00002302_00000001_roll_U_01

00002302 - Corresponde à identificação do indivíduo

roll - Corresponde ao modo de extração da impressão digital, podendo ser *roll* (resultado de quando se gira o dedo para abranger uma área maior) ou *slap* (equivalente a pressionar e soltar o dedo, sem realizar o movimento horizontal)

U - Nos casos em que a forma de extração é o *roll*, a impressão digital pode ser categorizada com U ou V. No caso do *slap*, a categoria é sempre a letra R.

Os demais elementos da nomenclatura que não foram especificados eram comuns à todas as imagens.

2. Literatura proposta

A implementação da solução foi dividida em duas partes: primeiro, o arquivo *project02.c* tem como objetivo gerar pares de impressões digitais em arquivos csv com ROIs otimizados para a suas classificações posteriores.

Baseando-se no MSPS (Multi-scale parameter search) descrito por Guilherme C.S. Ruppert et al.¹, esse método construído a partir de IFTs trabalha com uma imagem fonte e outra como destino buscando sobrepor a primeira ao espaço das coordenadas da segunda, ao mesmo tempo em que extrai a região central da impressão digital como a área de interesse para nossa análise.

Para a segunda parte da solução, os pares de imagens obtidos na primeira parte eram utilizados como dados para o treinamento e teste das redes neurais. Baseando-se nas redes de aprendizado contrastivo, a rede neural desenvolvida permite que os pares de impressões digitais sejam classificados a partir da similaridade por cosseno dos vetores de atributos, utilizando a loss e a acurácia para avaliar se a construção do dataset, da arquitetura e a escolha dos parâmetros são satisfatórias.

3. MSPS

3.1. Algoritmo

Para termos uma rede neural mais eficiente na identificação da biometria de indivíduos, é necessário realizar o processamento de imagens no banco de dados a ser utilizados, alinhando as imagens da melhor maneira possível.

Ao utilizar o algoritmo de MSPS descrito no arquivo *project02.c* é necessário trabalharmos com alguns parâmetros que definem o gradiente de transformação que será aplicado às imagens na tentativa de encontrar a melhor sobreposição da fonte nas coordenadas do destino.

Após executado o algoritmo, é gerado um arquivo csv com os pares de imagens e também uma pasta com o ROI reposicionados extraídos das imagens. Essas informações serão utilizadas como dados da rede neural.

3.2. Parâmetros

O arquivo *deltas.txt* possui a descrição de 5 valores para 4 parâmetros diferentes. Cada linha representa um desses parâmetros e os valores são as variações que eles assumem dentro do programa para, então, ser realizada a sua otimização.

Parâmetros em *deltas.txt*

- Translação em x
- Translação em y
- Rotação
- Escala

A partir das variações dos valores desses parâmetros, foram criados 4 sets para testes, cujos valores podem ser observados nas tabelas abaixo:

SET1 - Padrão

4	5
1.00	3.00 5.00 20.0 50.0
1.00	3.00 5.00 20.0 50.0
0.50	1.00 5.00 30.0 60.0
0.01	0.03 0.05 0.07 0.10

SET2

Menor movimentação nos eixos,
menor intensidade de rotação,
menor possibilidade de expansão (escala)

4	5
0.50	1.00 1.50 3.00 5.50
0.50	1.00 1.50 3.00 5.50
0.00	1.00 10.00 20.0 30.0
0.01	0.03 0.05 0.07 0.10

SET3

Menor movimentação nos eixos,
maior intensidade de rotação,
maior possibilidade de expansão (escala)

4	5
0.50	1.00 1.50 3.00 5.50
0.50	1.00 1.50 3.00 5.50
0.00	5.00 10.00 25.0 45.0
0.02	0.03 0.07 0.10 0.30

SET4

Maior movimentação nos eixos,
maior intensidade de rotação,
maior possibilidade de expansão (escala)

4	5
1.00	5.00 20.00 50.0 90.0
1.00	5.00 20.00 50.0 90.0
0.50	10.00 30.0 60.0 90.0
0.02	0.03 0.07 0.10 0.30

Outra variável que também poderia alterar a qualidade do processamento da imagem é o tamanho do crop extraído no ROI. Assim, esse valor também foi testado múltiplas vezes para os diferentes deltas, resultando em 12 conjuntos de testes diferentes.

Tamanhos da ROI

75px
300px
500px

3.3. Testes realizados

Antes de realizarmos as mudanças de parâmetros, foi estabelecido os arquivos que seriam utilizados para testar as sobreposições. Ao todo foram escolhidas 12 combinações, sendo 6 pares correspondente ao mesmo indivíduo entre si e 6 pares entre indivíduos diferentes. Esses valores foram definidos visando diminuir a falta de balanceamento que os dados poderiam causar na rede neural.

Buscando ainda a melhor forma de equilíbrio, a combinação seguiu a seguinte tabela, em que os símbolos de iguais referem-se ao mesmo indivíduo e o símbolo diferente à pares com indivíduos diferentes. Vale ressaltar que nos casos da mesma pessoa, eram comparadas impressões digitais entre as diferentes formas de coleta (slap_R, roll_U ou roll_V)

Escolha dos arquivos em *compare.txt*

Fonte/ Destino	Roll U	Roll V	Slap
Roll U	≠	≠/=	,=
Roll V	,=	≠	≠/=
Slap	≠/=	,=	≠

Além das saídas já especificadas, o programa também gera um arquivo score demonstrando as distâncias obtidas na combinação da posição das imagens.

Essas informações foram extraídas de cada teste realizado e através de um algoritmo auxiliar, *gaussian.py*, foram gerados gráficos comparando as alterações causadas apenas pelo tamanho do frame e comparando todas

Arquivos escolhidos para testes

```
12
00002649_00000001_slap_S_01.png 00002649_00000001_roll_V_01.png
00002330_00000001_roll_V_01.png 00002330_00000001_roll_U_01.png
00002312_00000001_roll_U_01.png 00002312_00000001_slap_R_01.png
00002328_00000001_slap_R_01.png 00002328_00000001_roll_U_01.png
00002394_00000001_roll_U_01.png 00002394_00000001_roll_V_01.png
00002462_00000001_roll_V_01.png 00002462_00000001_slap_R_01.png
00002649_00000001_roll_V_01.png 00002312_00000001_roll_V_01.png
00002649_00000001_roll_U_01.png 00002330_00000001_roll_U_01.png
00002330_00000001_slap_R_01.png 00002312_00000001_slap_R_01.png
00002330_00000001_roll_U_01.png 00002649_00000001_roll_V_01.png
00002312_00000001_roll_V_01.png 00002649_00000001_slap_S_01.png
00002649_00000001_slap_S_01.png 00002312_00000001_roll_U_01.png
```

as curvas entre si para selecionar a melhor combinação de parâmetros a ser utilizada na rede neural.

3.4. Execução e resultados

Para executar o algoritmo *project02.c* é necessário ter instalado a biblioteca IFT em um ambiente linux. Após ter cumprido os requisitos, os seguintes comandos irão gerar os arquivos apresentados nesse relatório.

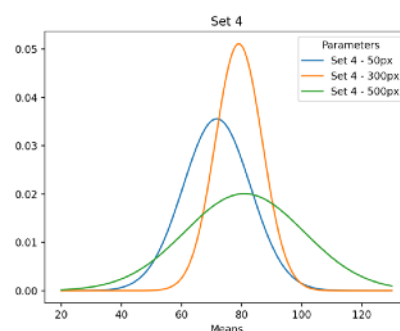
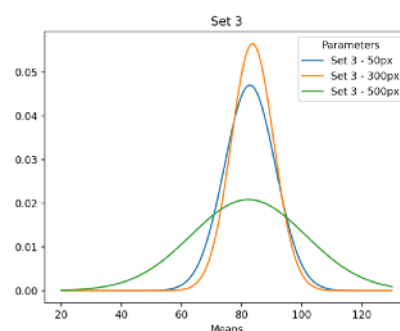
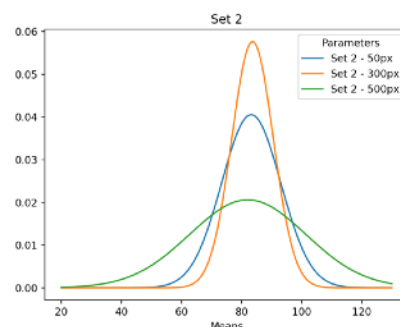
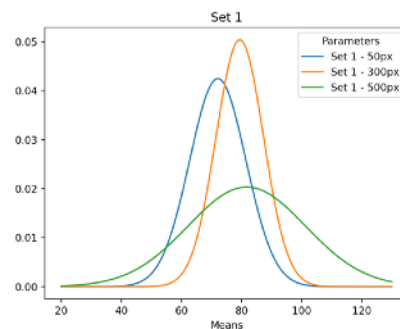
```
make project02
```

```
./bin/project02 images_02 compare.txt
deltas.txt pasta_destino
```

Ao observarmos os gráficos comparando os mesmos parâmetros delta com diferentes tamanhos do frame, podemos observar que o valor médio mantém-se parecido mas com uma variância muito maior quando o crop é maior, não sendo necessariamente benéfico utilizar áreas maiores em detrimento das menores.

Por sua vez, quando observamos o gráfico que compara todas as curvas, vemos uma vantagem na curva 'Set 2 - 300px', que possui o maior valor médio e menor variância entre todas as combinações, sendo esse set escolhido para ser utilizado na geração das imagens para a rede neural.

Assim, podemos concluir que variações menores nos parâmetros deltas produzem resultados melhores na combinação das imagens.



Média e variância dos scores encontrados para cada combinação de dimensão e set						
Set/Dimensão	50px		300px		500px	
	Média	Variância	Média	Variância	Média	Variância
Set1	72.06755556	9.3948074	79.37217778	7.91842433	81.689	19.61627856
Set2	83.17524444	9.84375864	83.62315556	6.9227696	82.01295556	19.37305545
Set3	82.74724444	8.48819416	83.60442222	7.0671197	82.22748889	19.15847379
Set4	71.72022222	11.21845243	79.04884444	7.81008417	80.8958	19.88374968

4.2. Rede Siamesa

A rede siamesa possui a seguinte arquitetura:

```

Bloco convolucional(1, 16)
  Conv2D
  BatchNorm
  ReLU
  MaxPool2D

Bloco convolucional(16, 128)
  Conv2D
  BatchNorm
  ReLU
  MaxPool2D

Flatten
Linear(128, 512)
ReLU
Linear(512, 256)
ReLU
Linear(256, 64)

```

Os dados para treinamento e testes foram divididos em 75% e 25% visando gerar números inteiros no dataset.

4.3. Execução e resultados

Além do conjunto de dados já apresentado que foi utilizado no treinamento da rede, foi gerado outro conjunto com o objetivo de testar a rede. Como no primeiro, a seleção destes foi balanceada entre pares de mesmos indivíduos e pares de indivíduos diferentes para que os resultados obtidos pela acurácia fossem o mais verosímil com o real desempenho do algoritmo.

No algoritmo, os caminhos para o arquivo csv e pasta de imagens para treinamento são:

```

./compare_set2_300.txt
./p2_registered_2_300

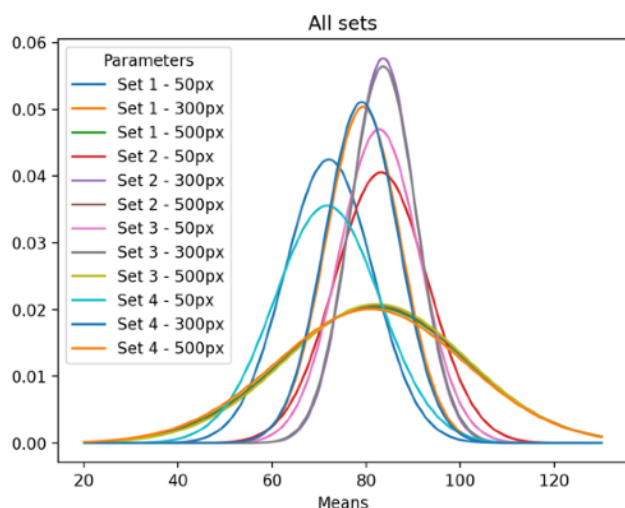
```

Os caminhos para o arquivo csv e pasta de imagem para teste da rede são:

```

./compare_test.txt
./p2_test

```



4. Aprendizado contrastivo

4.1. Algoritmo

Já realizado o processamento das imagens de impressões digitais, utilizamos esses dados em redes neurais para gerar modelos que identifiquem os indivíduos corretamente.

Ao utilizar o algoritmo fornecido das redes siamesas, buscamos melhorar os valores encontrados pela loss e pela acurácia. Em redes neurais, os dados utilizados geram grande influência nos resultados obtidos. Por isso o pré-processamento já realizado é considerado como crucial para o bom desempenho da rede.

A descrição completa do algoritmo utilizado pode ser encontrado no jupyter notebook *Redes Neurais*.

Arquivos escolhidos para testar rede

6
 00002332_00000001_slap_R_01.png 00002332_00000001_roll_U_01.png
 00002369_00000001_roll_V_01.png 00002369_00000001_roll_U_01.png
 00002375_00000001_roll_U_01.png 00002375_00000001_slap_R_01.png
 00002332_00000001_roll_U_01.png 00002369_00000001_roll_V_01.png
 00002369_00000001_roll_U_01.png 00002375_00000001_roll_U_01.png
 00002375_00000001_slap_R_01.png 00002332_00000001_roll_V_01.png

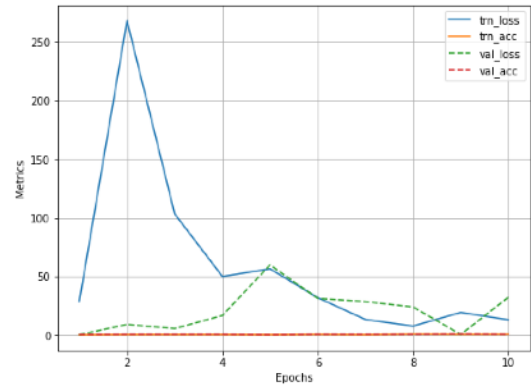
O objetivo dos testes era de gerar a menor loss com a maior acurácia possível. A quantidade de épocas executadas e o valor de `contrastive_thres` foram os responsáveis pelas maiores variações nos resultados obtidos.

Quanto mais épocas eram executadas, menores eram os valores de loss gerados - tantos para casos positivos quanto para negativos. Foram testados valores de épocas entre 5 e 20 devido ao custo computacional, contudo, no artigo que descreve a rede siamesa para aprendizado contrastivo é possível observar sinais de melhoria nos resultados para valores de épocas que se aproximam de 200.

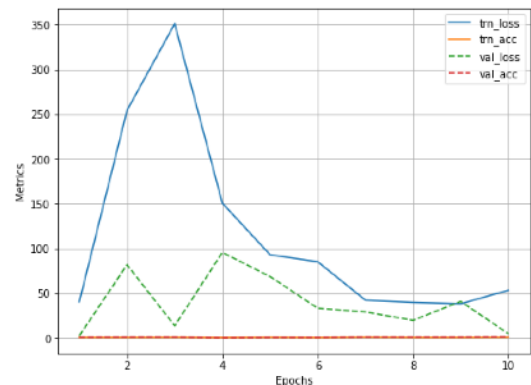
Os menores valores da loss impactam diretamente no valor escolhido para `contrastive_thres`, sendo necessário valores maiores para 5 épocas e menores para 20 épocas.

Em alguns dos testes, ao executarmos 5 épocas com o valor do `contrastive_thres` em 12, obtivemos 75% de acurácia. Os melhores resultados alcançados são apresentados na tabela a seguir.

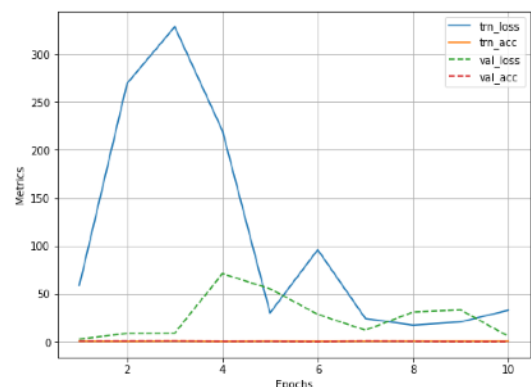
Valores gerados pela rede siamesa		
Epochs	contrastive_thres	Accuracy
5	12	0.75
10	9	0.83
10	12	0.66
20	7	0.66



Epochs = 10, Contrastive_thres = 12



Epochs = 10, Contrastive_thres = 9



Epochs = 10, Contrastive_thres = 7

5. Conclusão e trabalhos futuros

Os resultados apresentados nesse relatório, apesar de gerar algum sinal de convergência, não alcança grau suficiente de acurácia para ser utilizado enquanto solução. Os valores que poderiam ser considerados muito positivos não foram consistentes em testes posteriores.

As arquiteturas propostas são altamente capazes de produzirem melhores resultados quando observados mais atentamente os parâmetros utilizados. A continuação desse estudo segue na otimização desses parâmetros e da arquitetura rede - especialmente na extração de features da imagem.

Uma outra direção para trabalhos futuros é na geração de minúcias a partir das imagens de impressão digitais e calcular a distância entre esses vetores. Como discutido em sala, além de atingir o objetivo, o faria de maneira mais rápida e eficiente.

6. Referências

¹ Guilherme C.S. Ruppert, Giovani Chiachia, Felipe P.G. Bergo, Fernanda O. Favretto, Clarissa L. Yasuda, Anderson Rocha & Alexandre X. Falcão (2017) Medical image registration based on watershed transform from greyscale marker and multi-scale parameter search, Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization, 5:2, 138-156, DOI: 10.1080/21681163.2015.1029643

David A. Cardenas (2021) Constrastive Learning.