



universidade  
de aveiro

UNIVERSIDADE DE AVEIRO

TEORIA ALGORÍTMICA DA INFORMAÇÃO

---

## Projeto 3

RELATÓRIO FINAL

---

*Autores :*

84735 Pedro Ferreira  
84746 Rafael Teixeira  
84921 Rafael Direito

*Professores :*

Armando Pinho

7 de Janeiro de 2020

---

## Conteúdo

<b>1</b>	<b>Introdução</b>	<b>3</b>
<b>2</b>	<b>Objetivos</b>	<b>4</b>
<b>3</b>	<b>Normalized Information Distance</b>	<b>4</b>
<b>4</b>	<b>Normalized Compression Distance</b>	<b>4</b>
<b>5</b>	<b>Normalized Conditional Compression Distance</b>	<b>5</b>
<b>6</b>	<b>Seleção do Algoritmo de Reconhecimento Facial</b>	<b>7</b>
6.1	K-fold Cross Validation . . . . .	7
6.2	Classificação . . . . .	8
6.3	Obtenção do Erro Associado . . . . .	8
<b>7</b>	<b>Resultados</b>	<b>8</b>
7.1	NCD . . . . .	8
7.2	NCCD . . . . .	10
<b>8</b>	<b>Programa de Reconhecimento Facial</b>	<b>11</b>

---

## Lista de Tabelas

1	Erro associado à classificação das imagens sem pré-processamento. . .	8
2	Erro associado à classificação de imagens redimensionadas . . . . .	9
3	Erro associado à classificação de imagens quantizadas . . . . .	10
4	Erro associado à classificação das imagens. . . . .	11

## Lista de Figuras

1	Compressão das imagens. . . . .	6
2	Reconhecimento facial para 1 pessoa, usando a câmera do computador, com NCD e gzip. . . . .	13
3	Reconhecimento facial para 3 pessoas, usando a câmera do computador, com NCD e gzip. . . . .	13

---

# 1 Introdução

Inserido no plano curricular da disciplina de Teoria Algorítmica da Informação, do curso de Mestrado em Engenharia Informática, da Universidade de Aveiro e leccionada pelo professor Armando Pinho, este relatório é proveniente da execução do projeto III da cadeira.

O principal objetivo deste projeto é calcular a similaridade entre imagens, através do uso de NCD (Normalized Compression Distance) e NCCD (Normalized Conditional Compression Distance), ou seja, a semelhança entre imagens é calculada recorrendo apenas à compressão de dados.

Posto isto, este relatório tem como propósito expor o estudo realizado e ilustrar os resultados obtidos.

O relatório está organizado do seguinte modo: na Secção 2 estão definidos os objetivos para o projeto, na Secção 3 encontra-se uma explicação da Normalized Information Distance, na Secção 4 é realizada uma definição/explicação da NCD, assim como quais os compressores usados para o cálculo da NCD, na Secção 5 é efetuado um esclarecimento sobre a Normalized Conditional Compression Distance, na Secção 6 explica-se como foi realizada a seleção do algoritmo para o reconhecimento facial, na Secção 7 estão ilustrados os resultados obtidos com a concretização do estudo e na Secção 8 é especificada a implementação de um programa de reconhecimento facial, usando a câmara do computador

---

## 2 Objetivos

Como forma de orientação, os seguintes objetivos foram definidos para a concretização deste estudo:

- Compreender a NID, NCD e NCCD;
- Compreender e aplicar diferentes algoritmos de compressão para calcular a NCD;
- Compreender o algoritmo de compressão para calcular a NCCD;
- Avaliar os resultados obtidos na NCD e na NCCD, de forma a escolher o melhor mecanismo para o cálculo de similaridades entre imagens;
- Construir um programa de reconhecimento facial usando a câmera do computador.

## 3 Normalized Information Distance

A NID - Normalized Information Distance - é uma medida de distância universal entre objetos. Esta é baseada na complexidade de Kolmogorov, podendo ser obtida pela seguinte fórmula:

$$NID(x, y) = \frac{\max\{K(x|y), K(y|x)\}}{\max\{K(x), K(y)\}},$$

Uma vez que é baseada na complexidade de Kolmogorov, esta medida não é calculável, pelo que se criaram diversas aproximações que têm em conta o limite superior da mesma.

Dentro destas aproximações, surgem a NCD - Normalized Compression Distance - e a NCCD - Normalized Conditional Compression Distance -, objetos de estudo deste projeto.

## 4 Normalized Compression Distance

Como aproximação ao método da Normalized Information Distance, surge o mecanismo de Normalized Compression Distance. Esta distância pode ser calculada através da fórmula

$$NCD(x, y) = \frac{C(x, y) - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}},$$

---

onde  $C(x)$  corresponde ao número de bits necessário para um compressor representar  $x$  e  $C(x, y)$  representa o número de bits necessários para representar  $x$  e  $y$  concatenados.

O compressor acima referido pode ser, por exemplo, gzip, bzip2 ou lzma, entre outros. O compressor tem, contudo, que respeitar a seguinte regra:  $C(x, x) \approx C(x)$ .

No decorrer deste estudo, utilizámos os seguintes compressores para o cálculo da NCD:

- gzip (representativo dos algoritmos LZ)
- bzip2 (representativo dos algoritmos de compressão de blocos)
- lzma
- png
- jpeg

Relativamente aos valores da NCD, quanto mais próximos de 1 mais distantes os 2 elementos ( $x$  e  $y$ ) são. Em contraste, se a distância for próxima de 0, podemos afirmar que os 2 elementos são bastante semelhantes entre si.

Para o cálculo da Normalized Compression Distance, foi desenvolvido código Python, com recurso ao módulo Pillow. Inicialmente, o utilizador define um conjunto de imagens às quais pretende aplicar a NCD. Posteriormente, é recebido um par de imagens. Cada par de imagens vai ser concatenado, sendo que as 2 imagens vão gerar uma nova imagem, onde as imagens iniciais vão ser colocadas lado a lado (horizontalmente).

Após termos as 2 imagens originais e a imagem concatenada no mesmo formato, cada uma destas vai ser convertida para um array de bytes, se utilizarmos o algoritmo gzip, bzip2 ou lzma, sendo que este array irá ser comprimido por um destes algoritmos. Após termos os arrays de bytes comprimidos, basta aplicar a fórmula da NCD e obter o grau de similaridade entre as imagens. Se, por outro lado, utilizarmos o algoritmo png, ou jpeg, utilizamos diretamente as matrizes de valores greyscale representativas de cada imagem. Este mecanismo pode ser observado na Figura 1.

## 5 Normalized Conditional Compression Distance

A Normalized Conditional Compression Distance (NCCD) aparece como uma melhor aproximação à NID, uma vez que, ao recorrer a compressores condicionais, permite uma substituição direta do  $K$  por  $C$ . O compressor condicional utilizado foi uma adaptação para Python do compressor condicional fornecido pelo professor

---

```

# compress an image, given an algorithm
def compress_image(self, image):
    byteIO = io.BytesIO()
    if self.algorithm in ["png", "jpeg"]:
        image.save(byteIO, format=self.algorithm)
        return byteIO.getvalue()
    else:
        image.save(byteIO, format="PPM")
        if self.algorithm == "gzip":
            return gzip.compress(byteIO.getvalue())
        elif self.algorithm == "bzip2":
            return bz2.compress(byteIO.getvalue())
        elif self.algorithm == "lzma":
            return lzma.compress(byteIO.getvalue())

```

Figura 1: Compressão das imagens.

regente da cadeira, implementado em c++. Este utiliza modelos de contexto finito para realizar a compressão das imagens, sendo que esta é dividida em duas fases.

Na primeira fase, a imagem na qual vamos basear a compressão é processada, sendo gerados vários modelos estáticos de contexto finito. Estes modelos consistem na contagem dos diferentes níveis de cinzento, isto é, o número de vezes que cada um dos 256 valores de cinzento aparece.

Na segunda fase, a imagem a ser comprimida começa a ser processada com base nos modelos estáticos e, paralelamente, vai sendo gerado um segundo conjunto de modelos de contexto finito, estes dinâmicos, que acumulam os valores de cinzento da segunda imagem.

Estes dois conjuntos de modelos são usados para obter a probabilidade de um dado elemento, misturando as suas probabilidades individuais, ou seja, a probabilidade de um dado elemento é dada pela média pesada das probabilidades dos vários contextos.

$$P(x_{n+1}) = \sum_k P(x_{n+1}|x_{n-k+1..n}) * w_{k,n}$$

onde  $w_k$  identifica o peso atribuído ao modelo  $k$  e este vai sendo atualizado de acordo com a seguinte fórmula:

$$w_{k,n} = w_{k,n}^{gamma} * P(x_{1..n}|k)$$

---

onde *gamma* tem o valor de 0.99.

Depois de obtermos as probabilidades de ocorrência para todos os níveis de cinzento, calculamos o número de bits necessários para a representação do nível de cinzento que é feito com recurso a  $-\log_2(x)$ , onde  $x$  é a respetiva probabilidade pesada. O número de bits necessário para representar cada pixel da imagem é, então, somado, sendo o resultado desta soma o número de bits necessário para representar a toda a imagem, tendo como referência a imagem de base.

## 6 Seleção do Algoritmo de Reconhecimento Facial

Uma vez que contamos com várias opções para realizarmos o reconhecimento facial, decidimos que seria útil realizarmos, de uma forma sistemática, a comparação das mesmas, sendo que variamos, não só os algoritmos de compressão, como também as técnicas de pré-processamento. Como opções de algoritmos de compressão temos:

- NCD com gzip;
- NCD com bzip2;
- NCD com lzma;
- NCCD com um compressor de contexto finito e um contexto.

Como opções de pré-processamento temos:

- Quantização com 16 níveis e 64 níveis;
- Redimensionamento da imagem, com um rácio de [0.5, 0.25, 0.1].

### 6.1 K-fold Cross Validation

Para testarmos todas estas opções, contamos com um dataset composto por 400 imagens de 40 indivíduos diferentes, sendo que cada indivíduo tem 10 fotografias suas. Para a validação das diferentes opções decidimos recorrer a uma abordagem semelhante ao K-fold cross validation.

Este método é bastante conhecido na área de machine learning e consiste na divisão do dataset em  $k$  subsets ( $k=5$ , durante este estudo) e na utilização de  $k-1$  subsets para treino e 1 subset para teste, sendo que cada subset é utilizado uma vez para teste.

Como os nossos algoritmos são determinísticos e não necessitam de treino algum, apenas de imagens de referência, invertemos o k-fold, sendo que guardamos um dos subsets para referência e os 4 restantes para teste. O erro das 5 classificações



---

é guardado, sendo que a média dos mesmos é o erro final para aquele compressor combinado com o respetivo pré-processamento.

## 6.2 Classificação

Os compressores, por si só, não nos dão a informação sobre a quem corresponde a imagem. Esta informação é obtida através do resultado da NCD e da NCCD, sendo que assumimos que a NCD/NCCD mais baixo obtida quando comparando as 2 fotografias de cada uma das 10 pessoas usadas como referência, com a dada foto de teste, seria a pessoa a quem a foto corresponderia.

## 6.3 Obtenção do Erro Associado

O erro usado para determinarmos qual o melhor algoritmo de identificação foi obtido contando o número de classificações erradas e dividindo pelo total de classificações realizadas, ou seja, se  $n$  for o número de classificações incorretas, então o erro será de  $\frac{n}{320}$ .

# 7 Resultados

## 7.1 NCD

Após cálculo da NCD entre as imagens do dataset, obtiveram-se os resultados presentes na Tabela 1.

Com as Imagens Originais	
Algoritmo de Compressão Utilizado	Erro
gzip	0.256
bzip2	0.323
lzma	0.156
png	0.457
jpeg	0.959

Tabela 1: Erro associado à classificação das imagens sem pré-processamento.

Decidiu-se, também, testar como o redimensionamento e a quantização de uma imagem influenciam o erro de classificação. Para tal, testámos 3 rácios de redimensionamento da imagem e 2 níveis de quantização. Os resultados do estudo efetuado, podem ser observados na Tabela 2.

---

Com Variações na Resolução		
Rácio de Resolução	Algoritmo	Erro
0.5	gzip	0.262
0.5	bzip2	0.454
0.5	lzma	0.197
0.5	png	0.754
0.5	jpeg	0.975
0.25	gzip	0.481
0.25	bzip2	0.804
0.25	lzma	0.509
0.25	png	0.839
0.25	jpeg	0.977
0.1	gzip	0.781
0.1	bzip2	0.882
0.1	lzma	0.892
0.1	png	0.910
0.1	jpeg	0.972

Tabela 2: Erro associado à classificação de imagens redimensionadas

Comparando unicamente as classificações com recurso a imagens redimensionadas, podemos afirmar que quanto mais diminuimos o tamanho da imagem, maior vai ser o erro associado à sua classificação através da NCD. Apesar do aumento do erro de classificação, este redimensionamento acarreta uma poupança de recursos.

Na Tabela 3 podemos observar uma comparação do erro de classificação associado a imagens quantizadas.

Podemos afirmar que, com a diminuição dos níveis de quantização, aumenta o erro associado à classificação das imagens do dataset. Para além disto, pudemos observar que o intervalo temporal associado a estes testes foi muito superior aos demais, devido a ser necessário quantizar cada pixel da imagem. No entanto, através da quantização das imagens, surge também uma elevada poupança no que toca ao espaço ocupado por cada imagem.

De uma forma geral, podemos afirmar que os algoritmo que obtiveram melhores resultados foram o gzip e o lzma. Isto pode ser facilmente explicado com o facto de ambos os algoritmos recorrerem a compressão com base no algoritmo LZ77. O gzip, utiliza o algoritmo DEFLATE, que é uma combinação do algoritmo LZ77 e codificação de Huffman. Por outro lado, o lzma caracteriza-se por ser um versão me-

---

Com Quantização das Imagens		
Níveis de Quantização	Algoritmo	Erro
6	gzip	0.626
6	bzip2	0.801
6	lzma	0.732
6	png	0.438
6	jpeg	0.949
4	gzip	0.785
4	bzip2	0.937
4	lzma	0.595
4	png	0.627
4	jpeg	0.959

Tabela 3: Erro associado à classificação de imagens quantizadas

lhorada do algoritmo de compressão LZ77. Desta forma, compreende-se que tenham resultados semelhantes.

Podemos, também, afirmar que o algoritmo jpeg foi aquele que demonstrou um maior erro na classificação de imagens através da NCD. Este facto é devido ao jpeg ser um algoritmo de compressão com perdas, não verificando a condição necessária para a aplicação da NCD:  $C(x, x) \approx C(x)$ . Desta forma, não podemos considerar, de todo, os resultados associados a ao algoritmo jpeg.

O algoritmo de compressão png, apesar de ser um algoritmo bastante utilizado para compressão de imagens, obteve resultados muito abaixo do esperado, aquando comparado com os restantes algoritmos.

Tendo em conta os valores dos erros obtidos, podemos concluir que a NCD não é, de todo, a melhor opção para a classificação de imagens, uma vez que demonstra algumas falhas no reconhecimento de similaridades entre imagens em que os sujeitos estão orientados para a câmara de uma forma muitíssimo semelhante, com uma expressão semelhante e com uma aparência visual (cabelo, utilização de óculos, vestuário,...), também esta, semelhante. Caso alterássemos significativamente as poses dos sujeitos, o seu cabelo ou a utilização de óculos, provavelmente, a classificação iria estar associada a um erro bastante superior.

## 7.2 NCCD

Para a NCCD, devido a restrições de tempo (uma vez que o NCCD é um classificador muito lento) e de memória (a execução de compressões exige um consumo de memória bastante elevado), não realizámos K-fold cross validation, nem utilizámos

---

o dataset completo. Assim, realizámos a classificação de apenas dois sujeitos, sendo as primeiras duas fotos as referências e as restantes as fotos de teste.

Após cálculo da NCCD entre as imagens do dataset, obtiveram-se os resultados presentes na Tabela 4.

Com as Imagens Originais		
Pré-processamento	Contexto	Erro
Sem pré-processamento	ctx 1	0.563
Redimensionamento com um rácio de 0.5	ctx 1	0.563
Redimensionamento com um rácio de 0.25	ctx 1	0.438
Redimensionamento com um rácio de 0.1	ctx 1	0.5
Quantização com 6 bits - 64 níveis	ctx 1	0.188
Quantização com 4 bits - 16 níveis	ctx 1	0.063

Tabela 4: Erro associado à classificação das imagens.

Como podemos observar, o melhor erro foi obtido quando baixamos os níveis de quantização, isto deve-se ao NCCD usar modelos probabilísticos para determinar o custo de representação de uma imagem, ou seja, como vamos ter menos "buckets" de valores, existem mais repetições de contextos e de símbolos que ocorrem nesses mesmos contextos. Tal torna-se ainda mais notório para imagens semelhantes, sendo que a discrepância entre 2 imagens diferentes e 2 imagens parecidas é maior. Se diminuirmos muito mais os níveis de quantização, vamos começar a perder certos pormenores acabando por piorar outra vez a classificação.

Embora não seja analisado neste estudo, o algoritmo NCCD é o algoritmo mais lento que testámos, sendo que embora este tenha o erro de classificação mais baixo, se considerarmos imagens quantizadas, o seu uso torna-se inoportável, dado o tempo que este demora a obter uma resposta de classificação.

## 8 Programa de Reconhecimento Facial

Com o objetivo de aplicar o conhecimento adquirido na realização deste estudo, foi proposta a construção de um programa que, com a câmara do computador, fosse capaz de reconhecer uma pessoa. Assim sendo, foi construído um programa capaz de reconhecer uma pessoa em tempo real, comparando-a com imagens existentes, e de tirar fotografias, para o caso de não ter nenhuma fotografia para realizar o reconhecimento facial.

Posto isto, o programa faz uso da biblioteca OpenCV (open source), cuja as

---

funções, neste caso, são capturar vídeo através da câmara do computador e extrair todas as imagens capturadas durante a execução do programa. É, também, através desta biblioteca que a deteção facial é realizada. Uma vez que o reconhecimento facial será realizado através da NCD, é necessário transformar as imagens do formato do OpenCV, Mat, para o formato da biblioteca Pillow. O método de compressão usado para o reconhecimento facial foram o gzip e o lzma, pois, com base nos resultados obtidos, são os algoritmos de compressão que apresentam os melhores resultados. Por fim, sempre que o resultado da compressão for menor do que um certo limite, significa que reconheceu a(s) pessoa(s), e, conseqüentemente, o rectângulo vermelho associado a cada pessoa, usado para a deteção facial, muda para azul e escreve o nome da imagem por cima do rectângulo.

Após a fase de desenvolvimento, prosseguimos com a fase de testes, de forma a perceber quão rápido e eficaz era o reconhecimento facial com os diferentes compressões. Para a concretização dos testes, foram tiradas 4 fotografias a 3 pessoas, resultando num dataset de 12 imagens no formato png. Chegou-se à conclusão que se estiver apenas uma pessoa no raio de visão da câmara, obtemos uma boa classificação, Figura 2, mas se estiverem várias pessoas no raio de visão da câmara, a classificação torna-se dúbia, como podemos verificar na Figura 3. Isto deve ao facto da diferença dos resultados da NCD para duas pessoas diferentes ser mínima, o que faz com que o limite de decisão seja ínfimo. Em relação ao desempenho dos compressores, o gzip é ligeiramente mais rápido que o lzma. A NCCD não foi testada, pois é extremamente lenta de calcular.

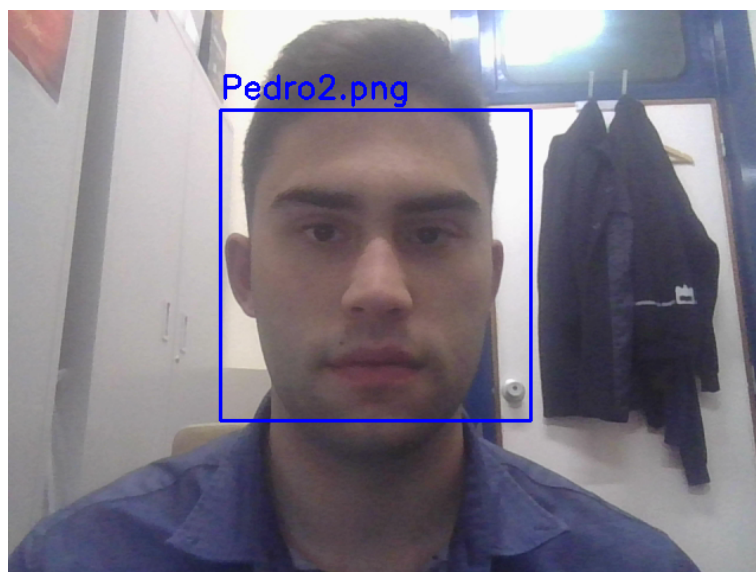


Figura 2: Reconhecimento facial para 1 pessoa, usando a câmera do computador, com NCD e gzip.

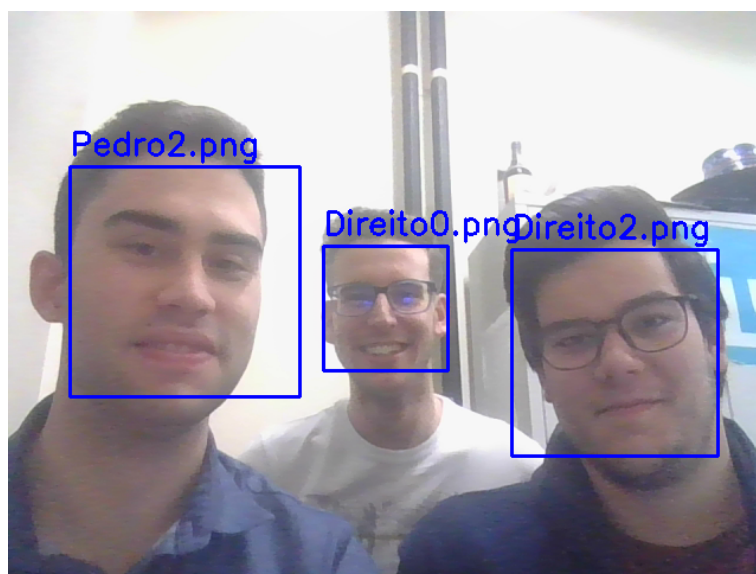


Figura 3: Reconhecimento facial para 3 pessoas, usando a câmera do computador, com NCD e gzip.