



Olhó-passarinho: uma extensão do TweeProfiles para fotografias

Ivo Filipe Valente Mota

VERSÃO DE TRABALHO

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Orientador: Luís Filipe Pinto de Almeida Teixeira (PhD)

Co-orientador: Carlos Manuel Milheiro de Oliveira Pinto Soares (PhD)

17 de Junho de 2014

Conteúdo

1	Introdução	1
1.1	Motivação	1
1.2	Objetivos	2
1.3	Estrutura do documento	2
2	Trabalhos relacionados	3
2.1	Clustering	3
2.1.1	Clustering por partição	4
2.1.2	Clustering hierárquico	5
2.1.3	Clustering baseado em densidade	7
2.1.4	Clustering baseado em grelhas	7
2.1.5	Funções de distância	8
2.2	TweeProfiles	12
2.3	Representação de imagens	12
2.3.1	Representação matricial	12
2.3.2	Histogramas	13
2.3.3	Descritores de cor	13
2.3.4	Descritores de textura	16
2.3.5	Descritores de forma	18
2.3.6	Descritores local	19
2.3.7	Descritores baseado em vocabulário visual	20
2.4	Trabalhos relacionados	21
3	Plano de trabalho e conclusões	23
3.1	Plano de trabalho	23
3.2	Ferramentas de apoio	24
3.3	Conclusão	25
	Referências	27

Lista de Figuras

2.1	Dendrograma [1]	6
2.2	Matriz confusão de dois objetos com atributos binários	10
2.3	Imagem em tons cinza e respetivo histograma	13
2.4	Sub-imagem e bloco de imagem. <i>Retirada de</i> [2]	17
2.5	Exemplo de formas de objetos que podem ser descritas eficazmente pelo descritor baseado em região. <i>Retirada de</i> [3]	18
2.6	Amostras de regiões com variância conjunta normalizadas de grupos que correspondem a uma única palavra visuais: (a,c,d) regiões SA - <i>Shape Adapted</i> ; (b) regiões MS - <i>Maximally Stable</i> . <i>Retirada de</i> [4]	20
3.1	Diagrama de Gantt com plano de trabalho	23

Abreviaturas e Símbolos

GoF/GOP	Grupo de <i>Frames</i> / Grupo de Imagens
SA	<i>Shape Adapted</i>
MS	<i>Maximally Stable</i>

Capítulo 1

Introdução

As redes sociais são uma excelente fonte de informação sempre em atualização, que fornece aos investigadores vasta quantidade e variedade de dados. Este dados apresentam-se de diferentes formas como textos, imagens e vídeos.

O Twitter é um serviço de microblogging, que permite aos utilizadores partilharem mensagens, designadas por *tweets*, até um máximo de 140 caracteres. O Twitter, ao contrário de outras redes sociais como o Facebook e LinkedIn que utilizam uma rede de comunicação bi-direcional, esta utiliza uma infraestrutura assimétrica onde existem os "*friends*" e os "*followers*". Supondo que é um utilizador do Twitter, os "*friends*" corresponde às contas das pessoas que o utilizador segue e os "*followers*" corresponde às contas das pessoas que o seguem [5].

O TweepProfiles [6], é uma ferramenta que recorre à rede social Twitter com o objetivo de identificar padrões em mensagens escritas em português partilhadas nesta rede social. Esta ferramenta utiliza processos de *Data Mining*, mais precisamente de *Text Mining*. A principal característica do TweepProfiles é o facto de utilizar a tarefa de *clustering* para identificar padrões, isto é, não se limita a apresentar *clusters* pelo conteúdo das mensagens (o texto), mas associando também as dimensões temporais e espaciais das mensagens.

1.1 Motivação

Para além do Twitter ser uma rede social que permite a partilha de *tweets*, este permite a partilha de imagens a partir do próprio serviço, ou através de outros serviços como Twitpic ou Flickr. Também, a partilha da ligação a uma imagem, é possível através do serviço Instagram, sendo que neste caso o acesso à imagem é redirecionado para o serviço Instagram. Devido ao grande número de utilizadores e de informação partilhada a todo o instante no Twitter, este torna-se um excelente serviço de pesquisa e análise, proporcionando aos investigadores e empresas uma quantidade e variedade de dados necessários para o desenvolvimento de ferramentas de extração de conhecimento e identificação de padrões.

1.2 Objetivos

Esta dissertação tem como principal objetivo a criação de uma extensão para o TweepProfiles através de técnicas de processamento de imagem e *Data Mining*, que permita a identificação de padrões em imagens partilhadas no serviço de microblogging Twitter, com representação de *clusters*.

Será assim necessário o desenvolvimento de um módulo responsável pela recolha das imagens através dos urls de tweets partilhados no Twitter. Para essa recolha será utilizada a plataforma TwitterEcho [7], que consiste num projeto open source, responsável por extrair e armazenar tweets de uma determinada comunidade de utilizadores, tendo sido desenvolvido com o intuito de ajudar os investigadores a terem facilidade de acesso a uma base de dados de tweets, na sua maioria, escritos na língua portuguesa. Posteriormente, deverá ser desenvolvido um módulo que utilize ferramentas de processamento de imagem para extração e análise da informação das imagens, e através da tarefa de *clustering*, seja capaz de apresentar *clusters* de imagens. Por fim, deverá ser integrado na ferramenta TweepProfiles, com objetivo de visualizar os *clusters* nas diferentes dimensões, como temporal, espacial e pelo conteúdo das imagens.

1.3 Estrutura do documento

O documento está organizado da seguinte forma: o capítulo 2 reflete o estado da arte, onde é apresentada uma pesquisa sobre os vários domínios científicos relacionados com as necessidades para o desenvolvimento do projeto de dissertação. Na secção 2.1 é descrito as características da tarefa de *clustering* do estudo e na secção 2.3, é referenciado as várias formas de representação computacional de imagens. Por fim, no capítulo 3, é referido o plano de trabalho para o desenvolvimento do projeto de dissertação, as ferramentas necessárias e as conclusões retiradas.

Capítulo 2

Trabalhos relacionados

Neste capítulo é apresentado o estudo preliminar realizado, tendo em vista a aquisição de competências e conhecimentos necessários para o desenvolvimento do projeto, que se focam essencialmente em análise de métodos de *data mining* e em técnicas aplicadas em visão por computador. Em primeiro lugar será exposto conteúdo relativamente a métodos de *clustering* como uma tarefa de *data mining*. Em seguida será apresentada formas de representação de imagens. Por fim, são referenciados alguns trabalhos relacionados com o projeto a desenvolver nesta dissertação.

2.1 Clustering

Extração de conhecimento em base de dados ou *Data Mining* é um processo de exploração de grande quantidades de dados que procura encontrar padrões interessantes [8]. Trata-se assim de uma fusão de estatística aplicada, sistemas de lógica, inteligência artificial, *Machine Learning* e gestão de base de dados [9]. Este processo é caracterizado por várias tarefas possíveis de ser aplicadas, dependendo do problema abordado, tais como [10]:

- Detecção de anomalias (outliers/ alterações/ desvios) - Identificação de registos de dados incomuns, podendo ser erros nos dados ou objetos interessantes que apresentam comportamento diferente dos restantes;
- Regras de associação - Procura relações entre variáveis que ocorram frequentemente;
- Classificação - É a tarefa de generalizar uma estrutura conhecida e aplicar a novos dados, sendo essencialmente utilizada em tarefas de previsão;
- Regressão - Tenta encontrar uma função que se modela os dados com o mínimo de erro;
- Resumo - Trata-se da representação mais compacta do conjunto de dados, que pode incluir visualização e descrição através de um relatório,
- *Clustering* - Tarefa de descobrir grupos em que os dados apresentam de alguma forma semelhanças, sem o uso de estruturas previamente conhecidas

Este projeto terá como principal tarefa a realização de *clustering*. Pode-se definir *clustering* como "um processo de agrupamento de um conjunto de objetos de dados em vários grupos ou *clusters*, de modo que os objetos dentro de um *cluster* apresentem alta similaridade, mas que sejam muito diferentes de objetos de outros *clusters*. Diferenças e semelhanças são avaliados com base nos valores de atributos que descrevem os objetos e muitas vezes envolvem medidas de distância" [8].

A realização de *clustering* é assim uma escolha lógica para a extração de padrões em dados não supervisionados e para o agrupamento de *tweets* pela sua semelhança em conteúdo, neste caso as imagens, e com a integração de outras dimensões, como o tempo e espaço.

O *clustering* faz parte de um conjunto de técnicas aplicadas na aprendizagem não supervisionada. Enquanto que na aprendizagem supervisionada, existe um conjunto de dados previamente analisados e rotulados que são usados para treinar um modelo para que seja capaz de encontrar relação entre os atributos desse dados com novos conjuntos dados, na aprendizagem não supervisionada, não são utilizados conjuntos de dados previamente analisados e rotulados. Assim o processo de descoberta de padrões nos dados apenas tem em conta os dados presentes, tentando organizar as instâncias em grupos semelhantes [11].

Nesta secção serão apresentadas as principais características e técnicas para aplicação da tarefa de *clustering*, tais como, *clustering* por partição, *clustering* hierárquico, *clustering* baseado em densidade, *clustering* baseado em grelhas, funções de distância para o cálculo da similaridade entre objetos e por fim, a avaliação de *clusters*.

2.1.1 Clustering por partição

A utilização de métodos baseados em partições é a forma mais simples e elementar de realizar análise por *clustering*, em que um conjunto de objetos é distribuído em vários grupos ou *clusters* mais pequenos. É assumido que o número de *clusters* é conhecido antes da realização da tarefa, sendo esse valor tomado como o ponto de partida para aplicação de métodos baseados em partição [8].

2.1.1.1 Algoritmo K-Means

O algoritmo *k-means* é o melhor algoritmo de *clustering* por partição e o mais utilizado devido à sua simplicidade e eficiência [11]. É apresentado como sendo um algoritmo de *clustering* por partição, pois este divide o conjunto de dados em partições mais pequenas, formando assim os *clusters*.

Inicialmente é necessário que o utilizador indique o valor de k e o algoritmo irá iterativamente dividir o conjunto de objetos em k -clusters diferentes, baseado em funções de distância [11] que são apresentadas na subsecção 2.1.5.

Cada *cluster* apresenta um centroide que é o representante do grupo, sendo o valor médio de todos os objetos (instâncias) pertencentes ao *cluster*. Este centroide é recalculado de forma

iterativa até que seja atingido o critério de paragem. A convergência ou critério de paragem pode ser um dos seguintes enumerados:

1. Não ocorre (ou ocorre um valor mínimo) de alterações dos objetos para diferentes *clusters*.
2. Não ocorre (ou ocorre um valor mínimo) de alterações dos centroides.
3. Diminuição mínima da **soma do erro quadrático** (SEQ),

$$SEQ = \sum_{j=1}^k \sum_{x \in C_j} dist(x, m_j)^2, \quad (2.1)$$

onde k é o número de *clusters* pretendidos, C_j é j -ésimo *cluster*, m_j é o centroide do *cluster* C_j e $dist(x, m_j)$ é a distância entre uma instância x e o centroide m_j .

Assim, "o algoritmo *k-means* pode ser usado em qualquer aplicação com um conjunto de dados onde a média pode ser definida e calculada" [11].

No **espaço euclidiano**, o valor médio (centroide) de um *cluster* é calculada da seguinte forma:

$$m_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i, \quad (2.2)$$

onde $|C_j|$ é o número de pontos (instâncias) no *cluster* C_j . A distância entre um ponto x_i a um centroide m_j é calculado da seguinte forma:

$$dist(x_i, m_j) = ||x_i - m_j|| = \sqrt{(x_{i1} - m_{j1})^2 + (x_{i2} - m_{j2})^2 + \dots + (x_{ir} - m_{jr})^2}. \quad (2.3)$$

O pseudo-código deste algoritmo é apresentado no algoritmo 1.

Algorithm 1 K-Means

```

1: procedure K-MEANS(k: clusters, D: conjunto de dados)
2:   escolher k objetos de D como centroides dos clusters iniciais;
3:   repeat
4:     (re) atribuir cada objeto a ao cluster ao qual o objeto é o mais similar;
5:     actualizar o centroide do cluster;
6:   until clusters sem alterações;
7:   return conjunto de k clusters;
8: end procedure

```

2.1.2 Clustering hierárquico

O *clustering* hierárquico é outra abordagem importante na tarefa de *clustering*. Os *clusters* são criados sobre a forma de uma sequência em árvore (dendrograma). Os objetos (instâncias) encontram-se no fundo do diagrama, enquanto que o conjunto de todos os objetos encontra-se no topo do diagrama. Cada nó que se encontra no interior do diagrama possui nós filhos, sendo que cada nó representa um *cluster*. Assim designa-se por *clusters* irmãos, aqueles que derivam

de um mesmo *cluster*, isto é, do nó parente [11]. A figura 2.1 exemplifica a representação de um dendrograma.

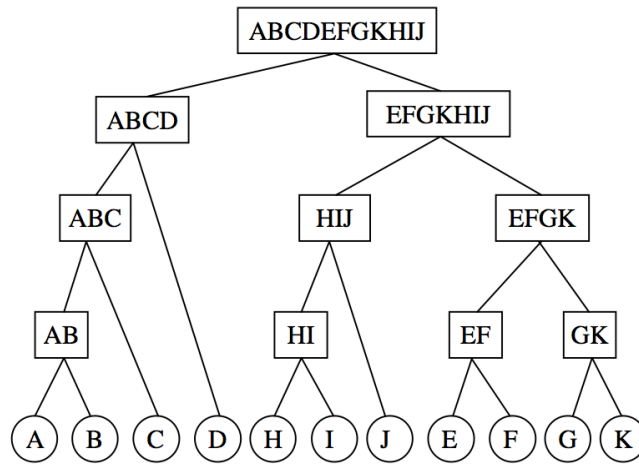


Figura 2.1: Dendrograma [1]

Existem dois tipos principais de métodos de *clustering* hierárquico, sendo eles [11] :

Clustering por aglomeração: O dendrograma é construído do nível mais baixo até ao mais alto, juntando sucessivamente e iterativamente os *clusters* com maiores semelhanças até existir um único *cluster* com todo o conjunto dos dados.

Clustering por divisão: O dendrograma é construído do nível mais alto até ao nível mais baixo, onde o processo tem início com um único *cluster* que possui todos os objetos, sendo dividido sucessivamente em *clusters* mais pequenos, até que estes sejam constituídos apenas por um único objeto.

Ao contrário do algoritmo *k-means*, que apenas calcula a distância entre os centroides de cada grupo ou *cluster*, no *clustering* hierárquico pode ser usado os vários métodos apresentados em seguida para determinar a distância entre dois *clusters* [11]:

Método Single-Link: Neste método, a distância entre dois *clusters* é determinada pela distância entre os dois objetos mais próximos (vizinhos mais próximo) pertencentes a *clusters* diferentes.

Método Complete-Link: Neste método, a distância entre dois *clusters* é determinada pela maior distância entre dois objetos (vizinhos mais distante).

Método Average-Link: Este método tenta manter um compromisso entre a sensibilidade a *outliers* do método *Complete-Link* e a sensibilidade do método *Single-Link* ao ruído existente nos dados. Para isso, é determinada a distância entre dois *clusters* através da distância média entre todos os pares de objetos nos dois *clusters*.

Método Ward: Este método, tenta minimizar a variância entre dois *clusters* unidos.

Assim conclui-se que o *clustering* hierárquico apresenta-se para determinados domínios, como bastante intuitivo para humanos, mas a interpretação dos resultados pode ser por vezes subjetiva. Outra característica interessante é o facto de, ao contrário do *clustering* por partição, no *clustering* hierárquico não ser necessário especificar logo à partida o número de *clusters*.

2.1.3 Clustering baseado em densidade

Os métodos de *clustering* por partição ou hierárquico estão preparados para encontrar *clusters* que apresentam formas geométricas circulares, sendo ineficiente quando as formas destes grupos são por exemplo elípticas. Assim, para descobrir *clusters* com formas arbitrárias, pode ser usado métodos baseados na densidade dos objetos [8]. Um dos algoritmos mais conhecidos que utiliza este tipo de método é o DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) que é capaz de encontrar *clusters* através da análise da densidade e proximidade dos objetos pertencentes a um conjunto de dados. Para esta análise é necessário previamente atribuir um valor que definirá o raio da vizinhança considerada para cada objeto. Esse parâmetro é designado por ϵ e terá de ser necessariamente maior que 0. Assim, a ϵ -vizinhança de um objeto x é o espaço dentro de um raio com valor ϵ , centrado em x [8]. Já para determinar a densidade de uma vizinhança, é utilizado o parâmetro *MinPts* também previamente definido, que especifica o número mínimo de objetos vizinhos que um objeto necessita ter em seu redor, para ser considerado como objeto central. Os passos necessários para a implementação deste método são apresentados no algoritmo 2

Uma das grandes vantagens do *clustering* baseado em densidade é que neste não é necessário uma previa definição do número de *clusters*, sendo apresentados os que encontrar consoante os dados que possui e os parâmetros definidos.

2.1.4 Clustering baseado em grelhas

Os métodos de *clustering* discutidos até agora, apresentam algoritmos que se adaptam a distribuição dos dados no espaço. Em alternativa, o *clustering* baseado em grelhas é orientado ao espaço, na medida em que divide o espaço em células independentemente da distribuição dos objetos de entrada. Este quantifica o espaço num número finito de células, que formam uma estrutura de grelhas sobre a qual são executadas as operações de *clustering*. Este método apresenta como principal vantagem, o tempo baixo de processamento, que normalmente é independentemente da quantidade de dados, no entanto, este depende do número de células em cada uma das dimensões no espaço quantizado [8].

O algoritmo STING (*Statistical Information Grid*) [12] é um dos algoritmos utilizados para *clustering* baseado em grelhas. Este divide o espaço em células retangulares, correspondente a diferentes resoluções que forma uma estrutura hierárquica, sendo a base o nível 1, os filhos o nível 2, e assim sucessivamente. Cada célula pertencente a um nível superior é dividida para formar células de menor dimensão no nível inferior seguinte. Assim, sabe-se que o nível mais baixo apresenta

Algorithm 2 DBSCAN

```

1: procedure DBSCAN(MinPts: limiar da vizinhança, D: conjunto de dados,  $\epsilon$  : raio )
2:   Marcar todos os objetos como não selecionados;
3:   repeat
4:     escolher aleatoriamente um objeto p não selecionado;
5:     atualizar objeto p como selecionado;
6:     if o  $\epsilon$ -vizinhança de p tem pelo menos MinPts objetos then
7:       criar um novo cluster C e adicionar objeto p ao cluster C;
8:       Seja N o conjunto de objetos na  $\epsilon$ -vizinhança de p;
9:       for cada ponto p' em N do
10:        if p' não selecionado then
11:          Marcar p' como selecionado
12:          if a  $\epsilon$ -vizinhança de p' tem pelo menos MinPts then
13:            adicionar ponto a N
14:          end if
15:        end if
16:        if p' não pertence a nenhum cluster then
17:          adicionar p' a C
18:        end if
19:      end for
20:      output C;
21:    else marcar p como ruído
22:    end if
23:  until todos os objetos selecionados;
24: end procedure

```

uma maior resolução. Isto permite que os *clusters* sejam encontrados recorrendo a uma pesquisa de cima para baixo (*clustering* por divisão, como explicado na sub-secção 2.1.2), passando por cada nível até atingir o mais baixo, retornando no fim as células mais relevantes para a consulta especificada. A informação estatística de cada célula é calculada e armazenada para o processamento de consultas futuras. Também é necessário ter em atenção que apenas é considerado para este algoritmo um espaço bidimensional. Um outro algoritmo com características semelhantes é o CLIQUE [13], que "identifica *clusters* densos em sub-espacos de máxima dimensão", isto é, são detetados todos os *clusters* em todos sub-espacos existentes e em que um ponto pode pertencer a vários *clusters* em sub-espacos diferentes.

2.1.5 Funções de distância

As funções de distância ou similaridade têm um papel fulcral em todos os algoritmos de *Clustering*. Existem enumeras funções de distância usadas para diferentes tipos de atributos (ou variáveis) [11]. Em seguida será apresentado diferentes funções distância para diferentes atributos como, numéricos, binários e nominal. Também será apresentado funções distância utilizados para as dimensões temporal e de conteúdo.

2.1.5.1 Atributos numéricos

As funções de distância mais utilizadas para variáveis numéricas são a **Distância Euclidiana** e **Distância Manhattan**. É utilizado $dist(x_i, x_j)$ para representar a distância entre duas instâncias de r dimensões. Ambas funções referidas anteriormente são casos especiais da função mais geral chamada **Distância Minkowski** [11]:

$$dist(x_i, x_j) = (|x_{i1} - x_{j1}|^h + |x_{i2} - x_{j2}|^h + \dots + |x_{ir} - x_{jr}|^h)^{\frac{1}{h}}, \quad (2.4)$$

onde h é um inteiro positivo.

Se $h=2$, temos a **Distância Euclidiana**,

$$dist(x_i, x_j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \dots + (x_{ir} - x_{jr})^2}. \quad (2.5)$$

Se $h=1$, temos a **Distância City-block (Manhattan)**,

$$dist(x_i, x_j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \dots + |x_{ir} - x_{jr}|. \quad (2.6)$$

Também, não menos importantes, são outras funções distância apresentadas em seguida:

Distância Euclidiana Ponderada : A ponderação é atribuída através de pesos pela importância que cada atributo representa relativamente a outros atributos.

$$dist(x_i, x_j) = \sqrt{w_1(x_{i1} - x_{j1})^2 + w_2(x_{i2} - x_{j2})^2 + \dots + w_r(x_{ir} - x_{jr})^2}. \quad (2.7)$$

Distância Euclidiana Quadrática : Trata-se de uma alteração da função **Distância Euclidiana**, elevando a mesma ao quadrado, o que faz com que seja progressivamente atribuído peso maior a pontos dos dados que estejam mais afastados.

$$dist(x_i, x_j) = (x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \dots + (x_{ir} - x_{jr})^2. \quad (2.8)$$

Distância Chebychev : Utilizada para casos em que há necessidade de definir dois pontos dos dados como diferentes, caso sejam diferentes em qualquer dimensão.

$$dist(x_i, x_j) = \max(|x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \dots + |x_{ir} - x_{jr}|). \quad (2.9)$$

2.1.5.2 Atributos binários e nominais

As funções apresentadas anterior apenas podem ser utilizadas com atributos do tipo numérico, assim será agora apresentado as funções de distância específicas para atributos do tipo binário e nominal.

Uma variável binária é aquela que apenas pode assumir dois estados ou valores, sendo normalmente representado pelo valor 0 e 1. Mas estes estados não apresentam um ordem definida.

Por exemplo, o caso de uma lâmpada, esta pode assumir apenas dois estados, ligado ou desligado, ou o gênero de uma pessoa, masculino ou feminino. Estes exemplos apresentam dois valores diferentes mas que não possuem qualquer ordem. As funções distância existentes para atributos binários são baseadas na proporção, sendo que a melhor maneira representar é através de uma matriz confusão [11].

		Objeto j	
		x = 1	x = 0
Objeto i	x = 1	a	b
	x = 0	c	d

$a+b+c+d = \text{número de variáveis}$

Figura 2.2: Matriz confusão de dois objetos com atributos binários

Os atributos binários ainda podem ser divididos em dois tipos de atributos diferentes, os simétricos e os assimétricos, sendo em seguida apresentado as funções distância para ambos os casos [11].

Atributos simétricos: Um atributo é simétrico quando ambos os estados (0 ou 1) têm a mesma importância e o mesmo peso, tal como ocorre no exemplo dado anterior com o atributo gênero (masculino e feminino). Para este caso, a função distância mais utilizada é designada por *simple matching distance*, que corresponde à proporção de incompatibilidade ou desacordo (equação 2.10).

$$dist(x_i, x_j) = \frac{b + c}{a + b + c + d} \quad (2.10)$$

Atributos assimétricos: Um atributo é assimétrico se um dos estados apresenta maior importância ou valor do que o outro. Normalmente o estado mais valioso é o que ocorre com menor frequência. No nosso caso iremos considerar o estado 1 como o mais valioso. Assim, a função distância mais frequentemente utilizada para atributos assimétricos é a *Jaccard distance*:

$$dist(x_i, x_j) = \frac{b + c}{a + b + c} \quad (2.11)$$

No caso de atributos nominais com mais de dois estados ou valores, a função distância mais utilizada, é baseada na *simple matching distance*. Dado dois objetos i e j , r corresponde ao número

total de atributos e o q ao número de valores que são mutuamente correspondidos entre os objetos i e j :

$$dist(x_i, x_j) = \frac{r + q}{r} \quad (2.12)$$

2.1.5.3 Dimensão Temporal

O tempo é representado apenas por uma dimensão, sendo que para calcular a distância, por exemplo, entre dois tweets t_i e t_j , apenas é necessário calcular a diferença dos tempos entre os mesmos. Supondo que os valores dos tempos são respetivamente Δ_i e Δ_j , o intervalo de tempo pode ser definido pela seguinte equação:

$$dist^T(t_i, t_j) = |\Delta_i - \Delta_j| \quad (2.13)$$

Sendo que, para a dimensão temporal, também é possível utilizar a função distância euclidiana (equação 2.5).

2.1.5.4 Dimensão Espacial

Ao contrário da dimensão temporal, a dimensão espacial apresenta mais do que uma dimensão, latitude e longitude. Estas apresentam-se sobre a forma numérica, sendo possível o cálculo da distância entre dois objetos através de funções de distância para atributos numéricos como referido anteriormente (2.1.5.1). Assim, para o cálculo entre pontos distribuídos num espaço poderá-se recorrer à função Minkowski (equação 2.4), à função Euclidiana (equação 2.5), à função Manhattan (equação 2.6) ou mesmo à função de Chebychev (equação 2.9), sendo que no caso mais específico de uma distribuição espacial geográfica, em que os pontos possuem latitude e longitude, é considerada mais apropriada a utilização da função distância Haversine (equação 2.14) pois esta toma em consideração a forma esférica da Terra. Obtendo um par de objetos x_i e x_j distanciados geograficamente, é considerado as latitudes ϕ_{x_i} e ϕ_{x_j} e longitudes λ_{x_i} e λ_{x_j} para determinar a distância através da seguinte equação,

$$dist^{Sp}(x_i, x_j) = 2R \sin^{-1} \left(\left[\sin^2\left(\frac{\phi_{x_i} - \phi_{x_j}}{2}\right) + \cos \phi_{x_i} \cos \phi_{x_j} \sin^2\left(\frac{\lambda_{x_i} - \lambda_{x_j}}{2}\right) \right]^{0.5} \right) \quad (2.14)$$

onde R representa o raio da Terra e que determina as unidades do resultado retornado pela função, sendo comum a utilização das unidades no sistema internacional (SI), o metro, podendo também ser representado em quilómetros devido ao fator de escala.

2.2 TweepProfiles

2.3 Representação de imagens

Na secção 2.1 foi apresentado o conceito e características da tarefa de *clustering* de uma forma geral. Nesta secção será exposto formas de representar imagens como dados. É importante salientar que a tarefa de *clustering* em imagem é muitas vezes associada à técnica de segmentação de imagem [14], não sendo este o objetivo deste projeto de dissertação, onde se pretende que para tarefa de *clustering* cada objeto seja considerado a imagem no seu conjunto.

2.3.1 Representação matricial

Uma imagem pode ser vista como um objeto (ou instância), sendo computacionalmente representada como uma matriz (um vetor bi-dimensional) de pixels. A matriz de pixels descreve assim a imagem como $N \times M$ m -bit pixels, onde N corresponde ao número de pontos ao longo do eixo horizontal, M o número de pontos ao longo do eixo vertical e m o número de bits por pixel que controla os níveis de brilho. Com m bits temos uma gama de valores para o brilho de 2^m , que varia entre 0 e $2^m - 1$. Assim se o valor de m for 8, os valores de brilho de cada pixel de uma imagem podem variar entre 0 e 255, que normalmente correspondem ao preto e branco respetivamente, sendo que os valores intermédios correspondem ao tons de cinza [15].

No caso de imagens a cores, o princípio é idêntico, no entanto ao invés de se usar apenas um plano, as imagens a cores são representadas por 3 componentes de intensidade, designado por modelo *RGB*, a que corresponde respetivamente às cores vermelho (Red), verde (Green) e azul (Blue). Para além deste esquema de cores, também existe outros como o *CMYK* composto pelas componentes de cor, azul turquesa, magenta, amarelo e preto. Usando qualquer esquema de cores, existem 2 métodos principais para representar a cor do pixel. No primeiro método é utilizado um valor inteiro para cada pixel, sendo esse valor como um índice para uma tabela, também conhecida como paleta da imagem, com a correspondência à intensidade de cada componente de cor. Este método tem como vantagem o facto de ser eficiente na utilização da memória, pois apenas é guardado um plano da imagem (os índices) e a paleta (tabela). Por outro lado, tem como desvantagem o facto de normalmente ser usado um conjunto reduzido de cores o que provoca uma redução da qualidade da imagem. Já o segundo método consiste na utilização de vários planos da imagem para armazenar a componente de cor de cada pixel. Este representa a imagem com mais precisão pois considera muito mais cores. Formato mais usual é 8 bits para cada uma das 3 componentes, no caso do *RGB*. Assim, são utilizados 24 bit para representar a cor de cada pixels, o que permite que uma imagem possa conter mais de 16 milhões de cores simultaneamente. Como era de esperar isto envolve um custo grande na utilização de memória, mas a constante redução do custo das memórias faz com que este seja uma boa alternativa à apresentada anteriormente [15].

2.3.2 Histogramas

Outras das formas de representar a informação de uma imagem é através de um histograma. Um histograma de uma imagem apresenta a frequência de ocorrência de níveis individuais de brilho, representado através um gráfico que mostra o número de pixels da imagens com um determinado nível de brilho. No caso de pixels representados por 8-bit, o brilho vai variar de 0 (preto) até 255 (branco) [15]. Também pode ser apresentado informação de cor sobre uma imagem através de um histograma, sendo para isso necessário apresentar 3 histogramas diferenciados, um para cada componente de cor, no caso do esquema RGB. A figura 2.3 apresenta um exemplo de um histograma de uma imagem com tons cinza, onde são representados o número de pixels para cada nível diferente de cinzento.

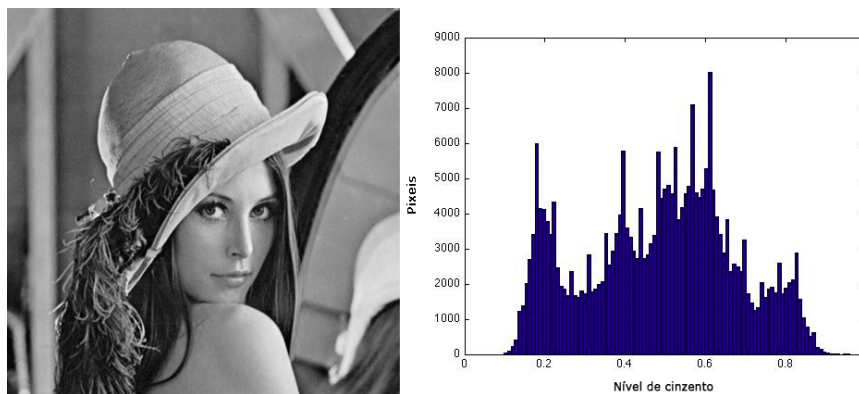


Figura 2.3: Imagem em tons cinza e respetivo histograma

2.3.3 Descritores de cor

A cor apresenta-se como um importante atributo da imagem para o olho humano e processamento por computador. Nesta subsecção são apresentados vários descritores de cor, utilizados para extração de informação e reconhecimento de similaridade em imagens. Por exemplo, o histograma de cores, referido na subsecção anterior 2.3.2, é um dos descritores de cor mais utilizados para caracterizar a distribuição da cor de uma imagem, mas apresenta uma baixa eficiência. Assim, em seguida é apresentado descritores de cor considerados pelo MPEG-7 [16, 17, 18, 19], que apresentam eficiência superior aos histogramas de cor.

2.3.3.1 Espaços de cor

Nesta subsecção é apresentado os vários descritores de espaços de cor especificado no MPEG-7 [19]. Existe uma vasta seleção de espaços de cores, tais como, RGB, YCbCr, HSV, HMMS, Monocromático e Matriz linear de transformação com referência a RGB. Estes são usados por outro descritor de cor, mais especificamente, o descritor de dominância de cor que será falado posteriormente. É utilizado também, um sinalizador para indicar a referência a uma cor primária e de mapeamento de um valor de referência do branco padrão.

Em espaços de cor, as componentes de cor são definidas como entidades de valor contínuo, sendo que podem ser representadas por valores discretos através de uma quantização uniforme, em que é especificado um número de níveis de quantização para cada componente de cor no espaço de cor. A única exceção é o espaço de cor HMMD.

O espaço de cor RGB é um dos modelos referidos mais utilizados, que apresenta três componentes distintas, vermelho, verde e azul, tal como foi referido na subsecção 2.3.1. Neste modelo é utilizado a combinação das 3 cores primárias para representar as diferentes cores. O modelo YCbCr provém do padrão MPEG-1/2/4 [19] e é definido pela transformação linear do espaço de cor RGB como demonstrado na equação 2.15:

$$\begin{aligned} Y &= 0.299 \times R + 0.587 \times G + 0.114 \times B \\ Cb &= -0.169 \times R - 0.331 \times G + 0.500 \times B \\ Cr &= 0.500 \times R - 0.419 \times G - 0.081 \times B \end{aligned} \quad (2.15)$$

Para o espaço de cor Monocromáticas, é usado apenas a componente Y do modelo YCbCr.

O espaço de cor HSV apresenta uma especificação mais complexa, tendo sido desenvolvido para fornecer uma representação mais intuitiva e para se aproximar mais do sistema visual humano. A transformação do modelo RGB para o HSV não é linear, mas é reversível [16]. Uma das componentes é a matiz (H - *Hue*), que representa a componente de cor espectral dominante na sua forma mais pura, como o verde, amarelo, azul e vermelho. Ao ser adicionado branco a estas cores, muda-se a cor, sendo que, adicionando menos branco, mais saturada é a cor. A saturação (S - *Saturation*) é precisamente outra das componentes deste modelo. Por fim, o valor (V - *Value*) corresponde ao brilho de cor.

O espaço de cor HMMD (*Hue-Max-Min-Diff*) [16, 19] é mais recente, que é caracterizado pela componente matiz, tal como o modelo HSV, pelo *max* e *min*, que são respetivamente o máximo e mínimo entre os valores R, G e B. Para descrever este modelo, também é utilizado a componente *Diff*, que corresponde à diferença entre o *max* e *min*. Para representar este espaço de cor, apenas é necessário três dos quatro componentes referidos anteriormente, como por exemplo, *Hue*, *Max*, *Min* ou *Hue*, *Diff*, *Sum*, onde *Sum* pode ser definida pela equação 2.16.

$$Sum = \frac{Max + Min}{2} \quad (2.16)$$

2.3.3.2 Cor dominante

O descritor de cor dominante fornece uma compacta representação das cores de uma imagem ou da região da imagem. Este apresenta a distribuição das cores mais representativas na imagem. Ao contrário do descritor de cor por histograma, a especificação do descritor de cor dominante, as cores mais representativas são calculadas a partir de cada imagem, em vez de ser fixado no espaço de cor, permitindo assim, uma representação das cores mais exatas e compacta, presentes numa região de interesse.

O descritor de cor dominante pode ser definido como,

$$F = c_i, p_i, v_i, s, (i = 1, 2, \dots, N)$$

onde N é o número de cores dominantes. Cada valor c_i da cor dominante é um vetor de valores das componentes do espaço de cor correspondente (por exemplo, um vetor de 3 dimensões no espaço de cor RGB). O valor p_i é a fração de pixels na imagem ou região da imagem (normalizado para um valor entre 0 e 1) que corresponde à cor c_i , sendo $\sum_i p_i = 1$. O opcional v_i descreve a variação dos valores de cor dos pixels em um *cluster* em torno da cor representativa correspondente. Por fim, a coerência espacial s é um único número representa a homogeneidade espacial global das cores predominantes na imagem [19].

2.3.3.3 Cor escalável

O descritor de cor escalável, pode ser interpretado como um esquema de codificação base que recorre à transformada de Haar, onde é aplicada aos valores do histograma de cor no espaço de cor HSV (referido na subsecção 2.3.3.1). De uma forma mais específica, o descritor de cor escalável, extrai, normaliza e mapeia de forma não linear os valores do histograma, numa representação inteira a 4-bit, dando assim mais relevância a valores mais pequenos. A transformada de Haar, é assim aplicada aos valores inteiros a 4-bit através das barras do histograma.

A extração do descritor é realizada com computação um histograma de cor com 256 níveis no espaço de cor de HSV com a componente matiz (H) quantificada a 16 níveis, e a saturação (S) e o valor (V) quantificado cada um para 4 níveis [17].

A aplicação típica do descritor, inclui como por exemplo, a busca de similaridade numa base de dados com conteúdo multimédia e pesquisa em enormes base de dados.

2.3.3.4 Grupo de *Frames* / Grupo de Imagens

O descritor de cor Grupo de Frames / Grupo de Imagens (GoF/GOP) é sobretudo utilizado para a representação conjunta de cores para várias imagens ou várias *frames* de um segmento de vídeo, contíguas ou não contíguas. Este baseia-se em histogramas, que capturam de forma confiável o conteúdo da cor de várias imagens ou *frames* de vídeo. Normalmente para um grupo de *frames* ou imagens, é selecionado uma frame chave ou imagem chave, que representará as características relacionadas com o grupo. Os métodos são altamente dependentes da qualidade da seleção da amostra representativa, o que pode levar a resultados pouco fiáveis caso não seja bem executada [19].

A estrutura do descritor GoF / GoP é idêntica à do cor escalável, com a exceção do campo, agregação, que especifica como os pixels da cor de diferentes imagens/*frames* foram combinadas antes da extração do histograma de cor. Os valores possíveis são média, mediana e cruzamento.

Uma das aplicações deste descritor, é a pesquisa em conjuntos grandes de imagens, para encontrar *clusters* de imagens semelhantes através da cor, em que é utilizado a interseção de histogramas como medida de similaridade da cor. A interseção de histogramas é obtido calculando

o valor mínimo de cada barra de cor do histograma ao longo das *frames*/imagens e atribui esse valor às barras de cor do histograma resultante. A interseção encontra as cores mínimas comuns nas *frames*/imagens e, portanto, pode ser utilizado em aplicações que requerem a detecção de um elevado grau de correlação da cor [17].

2.3.3.5 Estrutura de cor

Este descritor é uma generalização do histograma de cores que, apresenta algumas características espaciais da distribuição de cores em uma imagem. Este tem a particularidade de, para além de apresentar o conteúdo da cor de forma semelhante a um histograma de cor, este também apresenta informações sobre a estrutura de uma imagem, sendo esta a característica diferenciadora deste descritor de cor. Em vez de considerar cada pixel separadamente, o descritor recorre a uma estrutura de 8x8 pexels que desliza sobre a imagem. Ao contrário do histograma de cor, este descritor consegue distinguir duas imagens em que uma determinada cor está presente em quantidades iguais, mas que apresenta uma estrutura num dos grupos de pixels 8x8 que possui uma cor diferente nas duas imagens. Os valores de cores são representadas no espaço de cor HMMD com cone duplo, sendo o o espaço quantificado de maneira não uniforme em 32, 64, 128 ou 256 níveis. Cada valor de amplitude de um nível é representado por um código de 8 bits. Este descritor apresenta um bom desempenho na tarefa de recuperação de imagens baseado na similaridade [20].

2.3.3.6 Disposição de cor

O descritor de disposição de cor, caracteriza a distribuição espacial de cor de uma imagem. Este usa um vetor de cores representativas de uma imagem, expressas no espaço de cor YCbCr. O tamanho do vetor é fixado em 8x8 elementos para garantir invariância da escala do descritor. As cores representativas da imagem podem ser seleccionadas de diversas maneiras, mas a mais simples é através do calculo média de cor do bloco de imagem correspondente. O descritor disposição de cor, pode ser usado para em pesquisa rápida de bases de dados de imagens [18].

2.3.4 Descritores de textura

A textura das imagens é uma característica visual importante, que tem muitas aplicações na recuperação, navegação e indexação de imagens. Existem três descritores de textura, referenciados na norma MPEG-7 [2]. Em seguida é apresentada uma pequena descrição dos mesmos.

2.3.4.1 Descritor de textura homogénea

O descritor de textura homogénea (HTD - *Homogeneous Texture Descriptor*) descreve a distribuição estatística da textura de uma imagem. Existem neste descritor 62 interfaces de recurso, sendo 2 no domínio espacial e 60 no domínio das frequências. No domínio espacial, é extraído a média e o desvio padrão de uma imagem. No domínio das frequências, o espaço é dividido em 30 canais, sendo calculado o valor energético e o valor o desvio de energia da resposta do filtro

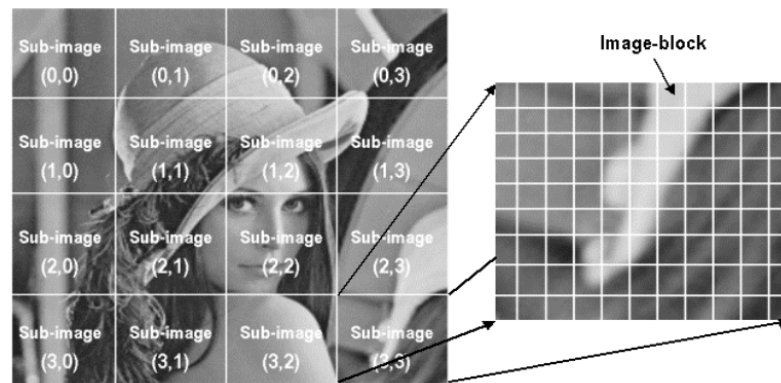


Figura 2.4: Sub-imagem e bloco de imagem. *Retirada de [2]*

Gabor em cada canal [2, 21]. Este desenho baseia-se no facto de a resposta do córtex visual possui banda limitada e no facto de o cérebro decompor o espectro em bandas na frequência espacial [2]. O descritor de textura homogénea é essencialmente utilizado em aplicações de recuperação de imagens por similaridade.

2.3.4.2 Descritor de histograma de borda

O descritor de histograma de borda (EHD - *Edge Histogram Descriptor*) apresenta-se sobre a forma de um histograma de 80 níveis, que representa a distribuição de borda local de uma imagem. Este descreve as bordas em cada sub-imagem. Estas sub-imagens são obtidas através da divisão da imagem numa grelha 4x4 como pode ser visto na figura 2.4. Existem 5 tipos de classificação diferentes das bordas de cada sub-imagem, sendo elas: vertical, horizontal, 45-graus, 135-graus e não direcional [2].

Este descritor é utilizado na recuperação de imagens, como por exemplo, imagens naturais ou de esboço, devido à sua textura homogénea. É também suportado por este descritor, a pesquisa baseada em blocos de imagem.

2.3.4.3 Descritor de textura homogénea

O descritor de navegação perceptual (HTD - *Perceptual Browsing Descriptor*) foi projetado para navegação em base de dados, mas principalmente para quando essa navegação necessita de recursos com sentido perceptual [2]. Este descritor é bastante compacto, que requer apenas 12 bits (máximo) para caracterizar regularidade (2 bits), direcionamento (3 bits x 2) e grosseirismo (2 bits x 2) da textura de uma imagem. A regularidade de uma textura pode apresentar valores numa escala entre 0 e 3, em que 0 indica uma textura irregular ou aleatória e 3 indica um padrão com direção e grosseirismo bem definidos. O direcionamento de uma textura é quantizada em 6 valores, variando de 0 a 150 em degraus de 30. Por fim, o grosseirismo de uma textura está relacionado com a escala e resolução de uma imagem. É quantizado em 4 níveis de 0 a 3, sendo 0 para um grão fino e 3 para uma textura grosseira. Estes valores têm uma relação com a divisão do espaço de frequência usado no cálculo no descritor de textura homogénea (HTD) [16]

2.3.5 Descritores de forma

O descritores de forma são dos descritores mais poderosos reconhecimento de objetos. Isto deve-se ao facto de os humanos serem exímios no reconhecimento de objetos característicos exclusivamente através da suas formas, provando que a forma muitas vezes possui informação semântica [3].

2.3.5.1 Descritor de forma baseado em região

O descritor de forma baseado em região (RSD - *Region-based Shape Descriptor*) [3] apresenta a distribuição de um pixel dentro de uma região de um objeto em 2 dimensões. Este permite a descrição de objetos simples, com ou sem buracos (figura 2.5), mas também permite a descrição de objetos mais complexos, que contém múltiplas regiões sem ligação.



Figura 2.5: Exemplo de formas de objetos que podem ser descritas eficazmente pelo descritor baseado em região. *Retirada de [3]* .

As principais características deste descritor são [3]:

- Fornece uma forma compacta e eficiente de descrever várias regiões disjuntas;
- Quando, no processo de segmentação de um objeto, ocorrem sub-regiões sem ligação, o objeto ainda pode ser recuperado, desde que a informação de quais as regiões foram divididas seja mantida e usada na extração do descritor;
- Apresenta uma boa robustez à segmentação de ruído.

2.3.5.2 Descritor de forma baseado no contorno

O descritor de forma baseado no contorno (CSD - *Contour-based Shape Descriptor*) [3] fundamenta-se na representação da curvatura espaço-escala (CSS - *Curvature Scale-Space*) do contorno. O contorno é uma propriedade importante na identificação de objetos semanticamente semelhantes. É também bastante eficiente em em aplicações onde a forma de objetos são muito variáveis, ou quando por exemplo, existem deformações de perspetiva, e apresenta um boa eficiência mesmo perante a existência de ruído nos contornos.

As principais características deste descritor são [3]:

- Consegue distinguir objetos que apresentem formas semelhantes mas que a forma do contorno apresenta propriedades bem diferenciadoras;
- Ele tem a capacidade de encontrar formas que são semanticamente similar para os seres humanos, mesmo quando existe uma significativa variabilidade intra-classe;
- É eficiente mesmo em casos de deformações não rígidas;
- É eficiente mesmo em casos de distorções do contorno devido a variações de perspectiva, sendo uma situação muito comum em imagens e vídeo.

2.3.6 Descritores local

Um descritor local permite a localização das estruturas locais de uma imagem de forma repetitiva. Estas são codificadas de modo a que sejam invariantes a transformações das imagens, tais como a translação, rotação, mudanças de escala ou deformações. Para a extração das características deste descritor é necessário seguir os seguintes passos:

- Encontrar um conjunto de pontos-chave;
- Definir uma região em torno de cada ponto-chave numa escala invariante;
- Extrair e caracterizar o conteúdo da região;
- Calcular o descritor da região normalizada;
- Combinar os descritores locais.

2.3.6.1 SIFT

O descritor SIFT (*Scale-invariant feature transform*) [22] é um descritor local que transforma uma imagem numa grande coleção de vetores de características locais invariantes a translação, rotação, mudanças de escala, e parcialmente invariante a mudanças de iluminação. Este descritor apresenta uma característica interessante, pelo facto de compartilhar uma série de propriedades em comum com as respostas dos neurónios do lobo temporal na visão dos primatas. Os pontos-chave SIFT derivadas de uma imagem, são usadas na indexação uma abordagem de vizinho mais próximo, para encontrar objetos candidatos. O modelo recorre a uma votação pela transformada Hough e a uma estimativa final pelo método dos mínimos quadrados. Quando 3 chaves estiverem em acordo, pode se afirmar que existe uma forte possibilidade da presença de um objeto.

Existe para além do descritor SIFT, outros descritores locais, tais como, o SURF (*Speeded Up Robust Features*), GLOH (*Gradient Location and Orientation Histogram*) e o HOG (*Histogram of Oriented Gradients*).

2.3.7 Descritores baseado em vocabulário visual

No reconhecimento de documentos de texto é utilizado o conceito de vocabulário textual, sendo muitas vezes designado por *bag of words* que em português significa, saco de palavras, onde existe um conjunto de palavras armazenadas pré definidas. Um texto é assim caracterizado, através da análise das palavras que possui, sendo contabilizado a frequência de palavras que estejam simultaneamente no texto e no *bag of words*, que assim atribuem um significado ao texto. É também utilizado uma lista de palavras que não acrescentam significado a expressões tais como, "o" ou "um", entre outras, que são removidas do texto durante a análise para não influenciarem os resultados. Um sistema de extração de informação de texto apresenta um número padrão de etapas [23].

Recentemente esta técnica foi adotada em aplicações de extração de informação em vídeo [24, 4]. Esta recorre a descritores locais invariantes para criar um vocabulário visual. A utilização de descritores locais como o SIFT (2.3.6.1) referido anteriormente, permite a criação de *clusters* de pequenas regiões imagens, selecionadas através de pontos-chave. Estas regiões, serão as "palavras" do vocabulário visual.

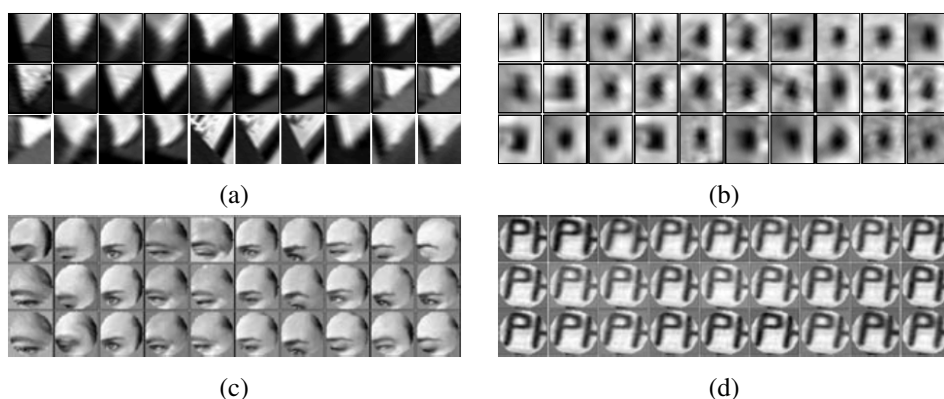


Figura 2.6: Amostras de regiões com variância conjunta normalizadas de grupos que correspondem a uma única palavra visuais: (a,c,d) regiões SA - *Shape Adapted*; (b) regiões MS - *Maximally Stable*. Retirada de [4]

Para a criação deste vocabulário visual, as regiões selecionadas podem ter em conta dois tipos diferentes de pontos de vista visual [24, 4]. Um é designado por *Shape Adapted* (SA), em que a forma adaptada sobre um ponto de interesse é elíptica. Este envolve um método iterativo para determinar o centro, escala e forma da elipse. O segundo, é designado por *Maximally Stable* (MS), que tem como característica o facto de as regiões selecionadas serem as que a área é aproximadamente estacionária enquanto o limiar de intensidade varia. Assim, as regiões SA tendem a centrar-se nos cantos e as regiões MS correspondem a formas de alto contraste em relação ao seu redor, como pode ser visualizado nos exemplos da figura 2.6.

Conclui-se assim que o objetivo é quantizar os descritores em *clusters* que serão as "palavras" visuais. O quantização vetorial pode ser realizado com recurso ao algoritmo de *clustering k-means* referido na secção 2.1.1, ou também a histogramas [24, 4].

Utilizado este tipo de descritor, é possível identificar imagens semelhantes, através da contagem de regiões de pontos-chave co-existent entre as imagens.

2.4 Trabalhos relacionados

A extração de informação em imagens é realizada já há vários anos, sendo muito utilizada na pesquisa de informação. Existem muitos trabalhos que, mesmo tendo objetivos diferentes, apresentam uma forte relação com o objetivo deste projeto de dissertação. Uma das áreas que se pode encontrar trabalhos relacionados é na detecção de objetos em vídeos [24, 4], onde se pretende identificar cenas de filmes semelhantes, tanto no mesmo filme, como entre filmes diferentes, recorrendo a descritores baseados em vocabulário visual. Como é de esperar, também existe vários trabalhos relacionados, que apresentam métodos diferentes para encontrar imagens semelhantes, como na pesquisa web [25, 26]. Na pesquisa bibliográfica realizada, foram encontrados dois trabalhos onde existe uma forte relação com o objetivo do projeto, e que apresentam técnicas que demonstraram bons resultados. O primeiro [27] utiliza os descritores baseados em vocabulário visual, tendo como diferenciação, a utilização de uma árvore hierárquica para reconhecimento de imagens semelhantes. O segundo [28], apresenta conjuntos de imagens semelhantes através da utilização do algoritmo de *clustering*, *k-means*, sendo que recorre a uma modificação que torna o algoritmo mais rápido, reduzindo o tempo de processamento mesmo com uma grande base de dados. Para a realização deste trabalho, foi utilizado histogramas de cor e histogramas preto e branco para descrever as imagens e ser possível assim identificar semelhanças entre elas.

Capítulo 3

Plano de trabalho e conclusões

Neste capítulo apresenta-se o plano de trabalho a ser realizado durante o desenvolvimento do projeto de dissertação através do diagrama de Gantt, com os períodos de tempo previstos para a sua execução, de acordo com os objetivos propostos no 1, secção 1.2. São também referenciadas as ferramentas e tecnologias necessárias para o desenvolvimento deste projeto e uma pequena conclusão.

3.1 Plano de trabalho

Nesta secção é apresentado na figura 3.1 o diagrama de Gantt com as tarefas previstas e com a respetiva distribuição das mesmas ao longo do tempo definido para o desenvolvimento do projeto de dissertação.

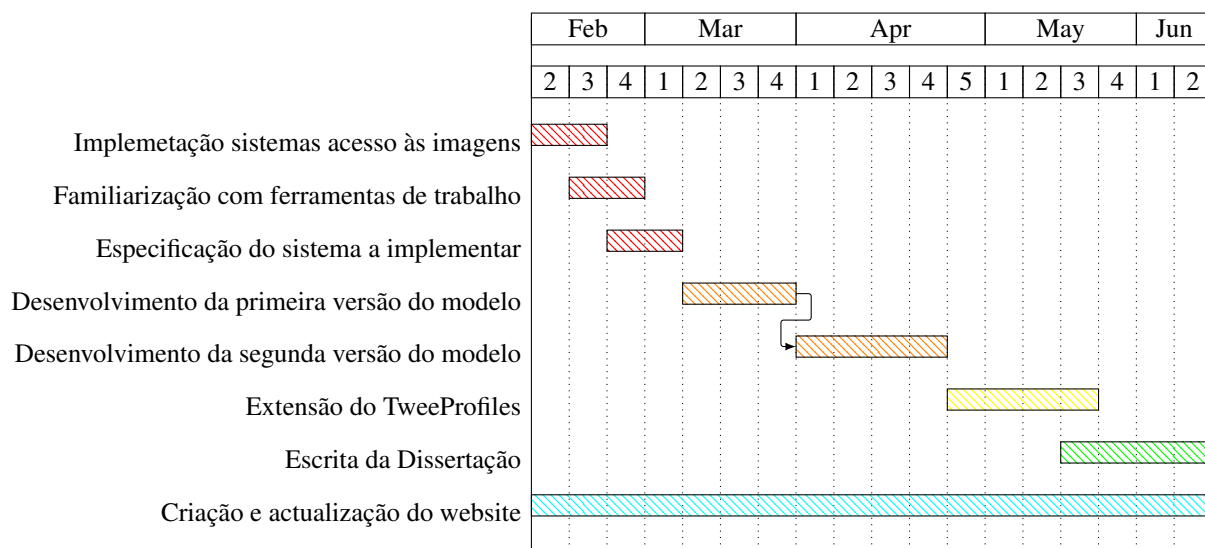


Figura 3.1: Diagrama de Gantt com plano de trabalho

Implementação sistemas acesso às imagens: Para a realização deste projeto será necessário realizar a recolha dos *urls* das imagens partilhadas através do serviço Twitter e o armazenamento das respetivas imagens para que possam posteriormente ser utilizadas para o objetivo desta dissertação referido no capítulo 1, secção 1.2. A duração prevista para esta etapa é de **duas semanas**.

Familiarização com ferramentas de trabalho: Nesta fase inicial, é necessário fazer uma avaliação e o estudo das ferramentas e do ambiente de desenvolvimento para as fases seguinte. A duração prevista para esta etapa é de **duas semana**.

Especificação do sistema a implementar: Em paralelo com a etapa anterior, dá-se início à especificação do sistema que irá analisar as imagens e ser capaz de apresentar *clusters* das mesmas. Será também definida a estratégia de ataque ao problema. A duração prevista para esta etapa é de cerca de **duas semanas**.

Desenvolvimento do sistema : Esta etapa encontra-se dividida em duas sub-etapas para a implementação do sistema a desenvolver. Enquanto que a segunda Esta etapa terá uma duração total prevista de **sete semanas**

1. **Primeira versão:** Que seja capaz de realizar a tarefa de *clustering* recorrendo a apenas algumas imagens. Esta Sub-tarefa terá uma duração prevista de **três semanas**
2. **Segunda versão:** Que seja capaz de realizar a tarefa de *clustering* recorrendo a uma base de dados maior. Esta Sub-tarefa terá uma duração prevista de **quatro semanas**

Extensão do TweepProfiles: Esta etapa refere-se ao processo de implementação do sistema desenvolvido na etapa anterior, em que se procederá à extensão da ferramenta TweepProfiles [6] com teste. Esta etapa tem uma duração prevista de cerca de **quatro semanas**.

Escrita da Dissertação: Após todos os testes realizados e feita a análise dos resultados obtidos, será realizado a última etapa, que corresponde à escrita do documento final. que deverá descrever todo o trabalho desenvolvido ao longo do semestre e as conclusões retiradas. Esta etapa tem uma duração prevista de cerca de **quatro semanas**.

Criação e atualização da página web: Esta será uma tarefa continua ao longo do desenvolvimento da dissertação, em que será criada uma pagina web com informação relativa ao projeto, e que será constantemente atualizada com os respetivos desenvolvimentos.

3.2 Ferramentas de apoio

Nesta secção faz-se referência às ferramentas e tecnologias utilizadas para o desenvolvimento do trabalho.

Todos os tweets estão alojados numa base de dados MongoDB. Para acesso à informação será utilizada a linguagem Python, mais especificamente, através da utilização da biblioteca Pymongo.

Será também necessário recorrer a uma base de dados relacional PostgreSQL para o armazenamento dos endereços de acesso à fonte das fotografias e de outras informações pertinentes que permita uma pesquisa facilmente filtrada.

Para o desenvolvimento do sistema de processamento das imagens será utilizado a ferramenta livre OpenCV.

3.3 Conclusão

Este relatório foi realizado com o intuito de preparar a realização do projeto de dissertação. Assim, primeiramente é apresentada uma introdução com a descrição do tema, das motivações e objetivos deste trabalho. De seguida, o capítulo 2 apresenta o estado da arte, onde foi realizada a revisão bibliográfica, recorrendo a pesquisa de livros e artigos científicos, informação esta que se revelou importante para a compreensão das tarefas necessárias a realizar, bem como dos conteúdos a dominar. Para além disto, este capítulo faz uma introdução teórica a alguns conteúdos que são necessários dominar. Por fim, foi apresentado o diagrama de Gantt que descreve as tarefas a realizar e a correspondente duração, e ainda, as ferramentas necessárias para o desenvolvimento do projeto.

Após a finalização deste relatório final, conclui-se que este projeto possui suporte científico e tecnológico para que seja concretizado.

Referências

- [1] Max Bramer. *Principles of Data Mining*. 2007.
- [2] Peng Wu, YM Ro, CS Won, e Yanglim Choi. Texture descriptors in MPEG-7. *Comput. Anal. Images ...*, páginas 21–28, 2001.
- [3] M. Bober. MPEG-7 visual shape descriptors. *IEEE Trans. Circuits Syst. Video Technol.*, 11(6):716–719, Junho 2001. doi:10.1109/76.927426.
- [4] Josef Sivic e Andrew Zisserman. Video Google: Efficient visual search of videos. *Toward. Categ. Object Recognit.*, 4170:127–144, 2006. doi:10.1007/11957959_7.
- [5] Matthew A Russell. *Mining the Social Web*, volume 54. 2011. doi:10.1081/E-ELIS3-120043522.
- [6] TDS Cunha. *TweeProfiles: detection of spatio-temporal patterns on Twitter*. Tese de doutoramento, Faculdade de Engenharia da Universidade do Porto, 2013. URL: http://paginas.fe.up.pt/~ei08142/files/mieic_en.pdf.
- [7] M Boanjak e Eduardo Oliveira. TwitterEcho: a distributed focused crawler to support open research with twitter data. *Proc. 21st ...*, 2012.
- [8] Jiawei Han, Micheline Kamber, e Jian Pei. *Data Mining, Second Edition: Concepts and Techniques*. 2006.
- [9] Dr. Matthew A North. *Data Mining for the Masses*. Global Text Project, 2012.
- [10] Usama Fayyad, Gregory Piatetsky-shapiro, e Padhraic Smyth. From Data Mining to Knowledge Discovery in Databases. páginas 37–54, 1996.
- [11] Bing Liu. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*. Springer, 2011.
- [12] Wei Wang, Jiong Yang, e Richard R. Muntz. STING: A Statistical Information Grid Approach to Spatial Data Mining. páginas 186–195, Agosto 1997.
- [13] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, e Prabhakar Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. *ACM SIGMOD Rec.*, 27(2):94–105, Junho 1998. doi:10.1145/276305.276314.
- [14] David A. Forsyth e Jean Ponce. *Computer Vision: A Modern Approach*. Pearson Education, Limited, 2011.
- [15] Mark S. Nixon e Alberto S. Aguado. *Feature Extraction and Image Processing*. 2002.

- [16] BS Manjunath e JR Ohm. Color and texture descriptors. *Circuits Syst. ...*, 11(6):703–715, 2001.
- [17] Charilaos Christopoulos, Daniel Berg, e Athanassios Skodras. The colour in the upcoming MPEG-7 standard. *Invit. Pap. Eur. ...*, páginas 1–4, 2000.
- [18] Leszek Cieplinski. MPEG-7 Color Descriptors and Their Applications. 7:11–20, 2001.
- [19] Leszek Cieplinski (mitsubishi Electric Ite-vil. The MPEG-7 Color Descriptors Jens-Rainer Ohm (RWTH Aachen, Institute of Communications Engineering).
- [20] Vinay Modi. Color descriptors from compressed images. 2008.
- [21] H Shao, J Ji, Y Kang, e H Zhao. Application Research of Homogeneous Texture Descriptor in Content-Based Image Retrieval. *Inf. Eng. ...*, (2008515):2–5, 2009.
- [22] D.G. Lowe. Object recognition from local scale-invariant features. *Proc. Seventh IEEE Int. Conf. Comput. Vis.*, páginas 1150–1157 vol.2, 1999. doi:10.1109/ICCV.1999.790410.
- [23] R Baeza-Yates e B Ribeiro-Neto. *Modern information retrieval*. 1999.
- [24] Josef Sivic e Andrew Zisserman. Video Google: A text retrieval approach to object matching in videos. *Comput. Vision, 2003. Proceedings. ...*, (Iccv):2–9, 2003.
- [25] Deng Cai, Xiaofei He, Zhiwei Li, Wei-Ying Ma, e Ji-Rong Wen. Hierarchical clustering of WWW image search results using visual, textual and link information. *Proc. 12th Annu. ACM Int. Conf. Multimed. - Multimed. '04*, 2004. doi:10.1145/1027527.1027747.
- [26] Bin Gao, Tie-Yan Liu, Tao Qin, Xin Zheng, Qian-Sheng Cheng, e Wei-Ying Ma. Web image clustering by consistent utilization of visual features and surrounding texts. Em *Proc. 13th Annu. ACM Int. Conf. Multimed. - Multimed. '05*, página 112, New York, New York, USA, Novembro 2005. ACM Press. doi:10.1145/1101149.1101167.
- [27] D. Nister e H. Stewenius. Scalable recognition with a vocabulary tree. ... *Vis. Pattern Recognition, 2006 ...*, 2, 2006.
- [28] Chuen-Horng Lin, Chun-Chieh Chen, Hsin-Lun Lee, e Jan-Ray Liao. Fast K-means algorithm based on a level histogram for image retrieval. *Expert Syst. Appl.*, 41(7):3276–3283, 2014.