

Universidade da Beira Interior
Departamento de Informática



Análise de Imagem: Medição de Área Edificada

José Manuel Pires da Costa, N°20095
Licenciatura em Engenharia Informática

Orientador do Projecto: **Prof. Doutor Hugo Proença**

Covilhã, Junho de 2010

Agradecimentos

A gratidão que sinto não passa pelo simples gesto de um obrigada, mas sim acima de tudo pelo demonstrar às pessoas que, com as suas acções, contribuíram de uma ou outra forma para que este trabalho fosse possível.

Para começar, gostaria de agradecer ao meu orientador de projecto, o Professor Doutor Hugo Proença, por me ter proporcionado a oportunidade de desenvolver este projecto. Obrigada também por todos os esclarecimentos e todo o conhecimento que me transmitiu de forma clara e precisa, que foram determinantes na realização deste projecto.

Agradeço também ao grupo *Soft Computing and Image Analysis Laboratory* (SOCIA Lab) pelas condições disponibilizadas para a realização deste projecto, bem como aos seus elementos pela ajuda prestada.

Quero agradecer à minha família, que sempre me apoiou em tudo ao longo da realização deste projecto e em toda a minha formação académica.

Por último agradeço à minha namorada, que me deu imensa força para que fosse possível a realização deste trabalho.

Obrigado!

Conteúdo

Agradecimentos	i
Conteúdo	iii
Lista de Tabelas	v
Lista de Figuras	vii
Lista de Algoritmos	ix
Acrónimos	xi
Glossário	xiii
1 Introdução	1
1.1 Motivação	1
1.2 Objectivos	2
1.3 Organização do relatório	3
2 Métodos de análise de imagens utilizando texturas	5
2.1 Extracção de características	5
2.1.1 Momentos Invariantes de Hu	6
2.1.2 Descritores de Texturas	8
2.2 Análise dos componentes principais	17
2.3 Redes Neurais Artificiais	21
3 Trabalho desenvolvido	25
3.1 Aquisição de Dados	25
3.2 Pre-Processamento	28

3.3	Extração de características	31
3.4	Normalização	36
3.5	Análise das componentes principais	36
3.6	Treino e classificação	37
3.7	Pós-processamento	39
4	Resultados	43
4.1	Resultados para o conjunto de treino	43
4.2	Resultados para o conjunto de teste	52
5	Conclusão e trabalho futuro	59
5.1	Conclusão	59
5.2	Trabalho Futuro	60
	Bibliografia	63

Lista de Tabelas

2.1	Matriz de uma imagem com níveis de cinza de 0 a 2	10
2.2	Matriz de co-ocorrência com 3 níveis de cinza.	10
2.3	Matriz de co-ocorrência para $\theta=0^\circ$	11
2.4	Matriz de co-ocorrência para $\theta=90^\circ$	11
2.5	Matriz de co-ocorrência para $\theta=45^\circ$	12
2.6	Matriz de co-ocorrência para $\theta=90^\circ$	12
2.7	Conjunto com dez amostras para as duas características . . .	18
2.8	subtracção da média em cada amostra	19
2.9	conjunto de novos componentes	21
4.1	Resultados das taxas de erro consoante as características seleccionadas pelo <i>Principal Component Analysis</i> (PCA) . .	45
4.2	Matriz Confusão da imagem 4.1(a) obtida com extracção de 40 características	46
4.3	Matriz Confusão da imagem 4.1(b) obtida com extracção de 30 características	47
4.4	Matriz Confusão da imagem 4.1(c) obtida com extracção de 40 características	47
4.5	Matriz Confusão da imagem 4.1(d) obtida com extracção de 40 características	48
4.6	Resultados das taxas de erro consoante a variação do nú- mero de neurónios da camada intermédia da rede neuronal	49
4.7	Matriz Confusão do melhor resultado da imagem 4.1(a) tendo em conta o segundo parâmetro	50
4.8	Matriz Confusão do melhor resultado da imagem 4.1(b) tendo em conta o segundo parâmetro	51
4.9	Matriz Confusão do melhor resultado da imagem 4.1(c) tendo em conta o segundo parâmetro	51

4.10 Matriz Confusão do melhor resultado da imagem 4.1(d)
tendo em conta o segundo parâmetro 51

Lista de Figuras

2.1	ângulos respeitantes ao cálculo da matriz co-ocorrência . . .	9
2.2	Imagem com uma textura periódica e o respectivo espectro .	16
2.3	Diagrama simplificado de uma Rede neuronal	22
2.4	Esquema representativo do método de aprendizagem supervisionado	23
3.1	Esquema das várias etapas do método proposto	26
3.2	Interface da aplicação Google Earth Slicer	27
3.3	Exemplo dos dois tipos de imagens utilizados	28
3.4	Imagem e respectiva máscara	29
3.5	Diferentes sistemas de cores	30
3.6	Exemplo do processo de percorrer uma imagem	32
3.7	Exemplo de um treino em uma rede neuronal	40
4.1	4 imagens do conjunto de treino	44
4.2	Gráfico das taxas de erro para várias características	45
4.3	Curvas <i>Receiver Operating Characteristic</i> (ROC) para os melhores resultados	48
4.4	Gráfico das taxas de erro para a variação do número de neurónios na camada intermédia	50
4.5	Curvas ROC dos melhores resultados da Tabela 4.6	52
4.6	3 imagens do conjunto de teste	53
4.7	Resultados obtidos para a imagem 4.6(a)	55
4.8	Resultados obtidos para a imagem 4.6(a)	55
4.9	Resultados obtidos para a imagem 4.6(a)	56
4.10	Resultados obtidos para a imagem 4.6(c)	57

Lista de Algoritmos

- 1 Este algoritmo é responsável por percorrer uma imagem e em cada iteração criar um filtro de tamanho n 33

Acrónimos

ANN	<i>Artificial Neural Network</i>
API	<i>Application Programming Interface</i>
DFT	<i>Discrete Fourier Transform</i>
FFT	<i>Fast Fourier Transform</i>
GLCM	<i>Grey Level CoOccurrence Matrix</i>
GPS	<i>Global Positioning System</i>
HSV	<i>Hue, Saturation, Value</i>
NAN	<i>Not a Number</i>
PCA	<i>Principal Component Analysis</i>
RGB	<i>Red, Green, Blue</i>
ROC	<i>Receiver Operating Characteristic</i>
SMA	<i>Segundo momento angular</i>
SOCIA Lab	<i>Soft Computing and Image Analysis Laboratory</i>

Glossário

Componentes HSV - sistema de cores formado pelas componentes hue (matiz), saturation (saturação) e value (valor);

Receiver Operating Characteristic (ROC) - ou simplesmente curva ROC é a representação gráfica da taxa de verdadeiros positivos com a taxa de falsos positivos.

Matriz de confusão - Matriz confusão é uma matriz $C \times C$, em que o valor de um item i, j representa a quantidade de casos na amostra que com resultado j foram determinados como i . Por Convenção a análise padrão é representada nas colunas.

Taxa da falsos positivos - Representa a taxa de regiões edificadas não detectadas como tal.

Taxa de verdadeiros positivos - Representa a taxa de regiões edificadas detectadas como tal.

Taxa de falsos negativos - Representa a taxa de regiões edificadas não detectadas.

Taxa de verdadeiros negativos - Representa a taxa de regiões não edificadas detectadas.

Capítulo 1

Introdução

Este projecto consiste no desenvolvimento de um sistema automático de análise e quantificação de áreas edificadas.

O principal objectivo baseia-se na construção de um agente automático que assinale as regiões ocupadas por construções humanas (edifícios, casas...) de uma determinada zona, com base nas características de textura extraídas de imagens de satélite obtidas através do Google Earth.

1.1 Motivação

A necessidade do processo de análise de imagens tem vindo a aumentar ao longo dos anos, principalmente quando este processo não consegue ser preciso quando efectuado pelo ser humano, ou depende de outros factores como, por exemplo, quando se trata de processos repetitivos. Por outro lado, o ser humano desde muito cedo que procura desenvolver tecnologia para o auxiliar e/ou facilitar nas tarefas do dia-a-dia, ou até mesmo substituí-lo.

Actualmente o interesse na área de processamento de imagens tem vindo a crescer, principalmente em aplicações como a melhoria da qualidade de imagens e um melhor processamento dos dados presentes nas imagens, proporcionando um aumento do seu uso nas mais diversas áreas.

Processar uma imagem subsiste em decompô-la com o objectivo de retirar a informação que nela está contida. É excessivamente complexo processar uma imagem como o sistema visual do ser humano, bem como realizar as mesmas tarefas que este, pois exige uma compreensão dos conhecimentos do ser humano. Esta característica torna o processamento de imagens uma área dependente do sistema ao qual esta se encontra associado, não havendo uma solução que permita resolver todos os problemas.

Dentro deste âmbito, a necessidade de um agente automático de reconhecimento de regiões edificadas torna-se importante, pois não existe actualmente nenhum mecanismo de recenseamento que permita saber o número de casas existente numa determinada região. Posto isto, o processo de análise das imagens para a detecção de certas regiões necessita de ser minucioso e bem definido para conseguir atingir os resultados pretendidos.

1.2 Objectivos

O objectivo deste projecto consiste no desenvolvimento de um agente automático de reconhecimento de regiões edificadas, para isso foi utilizada uma abordagem composta por várias etapas e para que os resultados finais fossem os pretendidos foi essencial que cada uma delas fosse implementada de maneira precisa.

A primeira etapa deste projecto consistiu na aquisição de dados. Para isso foi criada uma aplicação que permitisse extrair imagens satélite através da *Application Programming Interface* (API) do Google Earth.

A próxima etapa teve como finalidade a análise e a interpretação das imagens para adquirir informações regulares para se poder distinguir diferentes objectos de acordo com o que se pretende, sendo este processo baseado em texturas e na cor dos objectos existente nas imagens.

Neste trabalho, o processo de análise assentou essencialmente nas características dos níveis de cinza da imagem, nas componentes *Hue*, *Saturation*, *Value* (HSV), nos momentos da imagem e nos descritores de texturas, constituindo estas as características utilizadas para descrever as proprie-

dades das regiões que se pretende identificar.

De seguida foram seleccionadas as melhores características dentro destas, recorrendo á técnica de Analise das componentes principais (PCA) para eliminar redundância nos dados previamente obtidos.

A fase seguinte foi a fase da segmentação, onde foram criadas duas classes, a classe que representa o objecto que se pretende identificar e a classe que diz respeito ao resto da imagem.

Foi utilizado um classificador para treinar e simular as diferenças das regiões existentes nas imagens e assim obter um método automático de identificação de áreas edificadas, sendo as redes neuronais artificiais o classificador utilizado.

Obtidos os resultados da simulação do classificador foram calculadas taxas de erro para vários exemplos, bem como a matriz de confusão e da curva ROC, sendo que foram efectuados sucessivos testes em que houve a alteração de alguns parâmetros respeitantes as fases anteriores para se obter vários resultados até se chegar ao resultado final pretendido e também para se chegar a algumas conclusões no que diz respeito a alguns dos métodos utilizados.

Por último foi efectuada uma fase de pós-processamento que consistiu em aperfeiçoar as imagens resultantes obtidas do classificador.

1.3 Organização do relatório

Este relatório encontra-se dividido por capítulos e nesta secção vou descrever sucintamente o que consta em cada um deles. Neste capítulo, para além desta secção consta uma introdução onde se descrevem os objectivos, bem como a motivação que envolve o propósito deste projecto.

No capítulo 2, apresenta-se os principais fundamentos e métodos relacionados com a análise de imagens recorrendo às texturas, tais como a extracção de características, análise das componentes principais e por fim

os classificadores.

No capítulo 3, mostra-se o trabalho desenvolvido, desde a aquisição de dados até a classificação, passando pela extracção de descritores e pós-processamento.

No capítulo 4, demonstra-se num primeiro subcapítulo os resultados que dizem respeito ao conjunto de treino, mostrando a matriz confusão para vários casos bem como a curva ROC e num segundo subcapítulo os resultados obtidos a partir de imagens do conjunto de teste exibindo aqui as imagens "pintadas" a preto e branco como resultado final.

No capítulo 5, é feita uma conclusão do trabalho efectuado numa tentativa de retirar algumas deduções sobre a realização deste projecto, bem como discutir alguns pontos que possam ser acrescentados no futuro para se tentar obter ainda melhores resultados.

Capítulo 2

Métodos de análise de imagens utilizando texturas

Antes de realizar algumas das etapas do trabalho, foi necessário um estudo aprofundado de algumas abordagens e conceitos do processo de análise de imagens utilizando texturas, para uma melhor compreensão e uma melhor implementação desses métodos.

Neste capítulo serão expostos alguns desses conceitos que facilitaram o entendimento no desenvolvimento do trabalho.

2.1 Extracção de características

A extracção de características representa transformar os dados de entrada em características, isto é, quando os dados a serem processados são redundantes esses deverão ser transformados em um conjunto reduzido de características melhor representativo, este conjunto também pode ser denominado de vector de características.

No processo de análise de imagens normalmente utiliza-se um vector de características que figura uma representação mais concisa da imagem ou dos seus componentes, pois armazena os atributos mais expressivos das regiões da imagem. O número desses atributos determina o tamanho do vector de características, que usualmente depende das propriedades

que se pretende diferenciar.

Como iremos ver a seguir, normalmente tende-se a seleccionar os melhores atributos que melhor descrevem as propriedades dos componentes da imagem, reduzindo assim a dimensão do vector de características.

Nos atributos extraídos da imagem, de forma a auxiliar o processo de análise, destacam-se a cor, textura e forma.

De seguida explico alguns dos atributos, como os momentos invariantes de Hu e os atributos obtidos através dos descritores de texturas.

2.1.1 Momentos Invariantes de Hu

Em visão computacional o momento de uma imagem é uma certa média ponderada específica da intensidade dos pixéis da imagem e normalmente são escolhidos por conterem alguma propriedade atractiva. Estes podem ser aplicados a diversos aspectos de processamento de imagem que podem ir desde o reconhecimento de padrões invariantes até à codificação da imagem para representar uma estimativa. Dos vários momentos de uma imagem serão explicados mais detalhadamente os momentos invariantes de Hu.

Os momentos de Hu são invariantes à translação, à mudança de escala e também à rotação. Hu (1962) descreveu dois métodos diferentes para descrever os momentos invariantes[7].

O primeiro chamado de eixos principais, que no entanto se verificou que falhava quando as imagens não continham um eixo principal único, tais imagens são descritas como simétricas em relação à rotação.

O segundo método é o momento invariante absoluto. Hu derivou expressões das invariantes algébricas aplicadas ao momento da função gerada a partir de uma transformação de rotação. Estas consistem em grupos de expressões não lineares de momentos centralizados. O resultado é um conjunto de momentos invariantes ortogonais absolutos que podem

ser usados para a identificação de padrões invariantes à escala, posição e rotação. Foram utilizadas, numa experiência simples de reconhecimento de padrões, que identificou com sucesso vários tipos de caracteres. Estes são obtidos a partir dos momentos centralizados normalizados, os quais mostramos a seguir e posteriormente os sete momentos de Hu [13].

Definição de um momento discreto centralizado, como descreve HU (1962) [7]:

$$\mu_{pq} = \sum_{x=1}^N \sum_{y=1}^N (x - \bar{x})^p (y - \bar{y})^q P_{xy} \quad (2.1)$$

7 Momentos invariantes de Hu:

$$I_1 = \eta_{20} + \eta_{02}$$

$$I_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$$

$$I_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$$

$$I_4 = (\eta_{30} - \eta_{12})^2 + (\eta_{21} - \eta_{03})^2$$

$$I_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

$$I_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} - \eta_{03})^2 + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})]$$

$$I_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} - \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

2.1.2 Descritores de Texturas

As texturas possibilitam o aspecto visual de rugosidade ou suavidade de uma determinada superfície, proporcionando informações sobre a disposição espacial das mudanças da coloração de um objecto.

A textura é uma das características mais importantes na definição de uma imagem, e é caracterizada pela distribuição espacial dos níveis de cinza na vizinhança [12].

Shaban e Dikshit (1998) demonstram a importância da textura na segmentação de imagens e discutem a variação de determinados parâmetros nas suas experiências, alguns deles serão debatidos mais à frente [11].

A análise de textura proporciona o estudo das propriedades da superfície de um objecto para assim perceber como os seres humanos discriminam diferentes texturas, através de algoritmos que tenham capacidade de realizar esta tarefa.

Gonzalez e Woods (2002) descrevem a existência de três principais abordagens para a descrição de texturas: Abordagem estatística, abordagem espectral e abordagem estrutural [10]. De seguida explica-se as duas primeiras, pois estas são utilizadas no desenvolvimento do trabalho.

Abordagem estatística

A abordagem estatística é geralmente recomendada para texturas que não apresentam boa regularidade [3], onde a textura é descrita por um conjunto de medidas locais e por meio de propriedades não determinísticas que definem as distribuições dos níveis de cinza de uma imagem. De seguida apresenta-se a definição de matriz de co-ocorrência que serve para a extracção dos descritores de textura através desta abordagem, sendo depois apresentados alguns desses descritores.

Haralick (1973) descreve uma técnica para análise de textura utilizando matrizes de co-ocorrência (*Grey Level CoOccurrence Matrix (GLCM)*), para representar a distribuição da probabilidade da dependência espacial

da intensidade dos pixels [6].

Uma matriz de co-ocorrência é uma matriz que contém todas as combinações diferentes dos valores de intensidades dos pixels que ocorrem em uma imagem. Considera-se a relação entre dois pixels, em que um é intitulado pixel de referência e outro pixel vizinho. O segundo é o vizinho do primeiro e este pode ser o pixel à direita, à esquerda, acima e abaixo e ainda nas diagonais, e todos os pixels da imagem serão pixels referência desde o pixel existente no canto superior esquerdo até ao do inferior direito, sendo que é feita uma leitura na imagem desde um ao outro e para cada um deles existe oito pixels vizinhos ou até mais se considerar uma vizinhança maior.

A matriz de co-ocorrência de uma forma geral pode ser especificada de acordo com uma matriz $P(i, j, d, \alpha)$ na qual dois pixels vizinhos, separados por uma distância d e com orientação α ocorrem na imagem, sendo que um tem um nível de cinza i e outro j , isto é, i e j são os pixels vizinhos de acordo com a distância d , sendo a distância d analisada de acordo com a direção α . Os valores possíveis para α estão expostos na figura 2.1

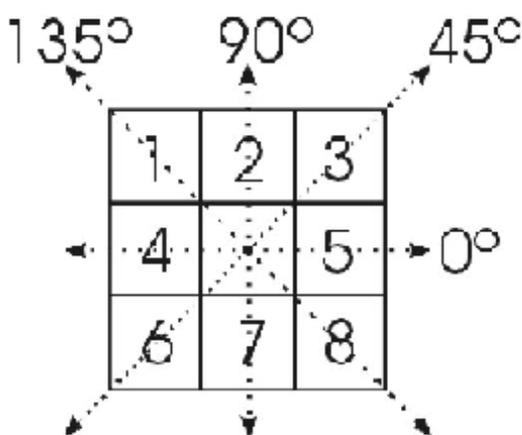


Figura 2.1: ângulos respeitantes ao cálculo da matriz co-ocorrência

E desta forma a matriz de co-ocorrência, que será obrigatoriamente quadrada, representa em cada elemento da matriz o número de vezes que

ocorreu uma transição de um nível de cinza i para j considerando a distância d entres estes mesmos pixéis na direcção α .

A construção destas matrizes é feita a partir das definições anteriores. Para construir esta matriz é necessária uma imagem com n níveis de cinza. De seguida apresenta-se um exemplo original de Gonzalez e Woods em que é considerada uma imagem com os valores dos pixéis representados na matriz 2.1 [10].

0	0	0	1	2
1	1	0	1	1
2	2	1	0	0
1	1	0	2	0
0	0	1	0	1

Tabela 2.1: Matriz de uma imagem com níveis de cinza de 0 a 2

Como de pode verificar esta imagem contém 3 níveis de cinza, $n=0$, $n=1$, $n=2$; o próximo passo é construir uma matriz quadrada que contém os níveis de cinza nas linhas e nas colunas, como demonstrado a seguir na figura 2.2

	0	1	2
0			
1			
2			

Tabela 2.2: Matriz de co-ocorrência com 3 níveis de cinza.

Para cada imagem haverá sempre quatro matrizes de co-ocorrência, uma para cada direcção, ou seja, uma para $\alpha=0^\circ$, outra para $\alpha=90^\circ$, outra para $\alpha=45^\circ$ e para $\alpha=135^\circ$. Por exemplo para uma distância $d=1$, e com $\alpha=0^\circ$ tem-se que verificar os pixéis imediatamente à esquerda e à direita.

Posto isto, cada posição da matriz de co-ocorrência conterà as variações dos níveis de cinza conforme os parâmetros já indicados e considerando os

índices da matriz. Por exemplo, a posição (0,0) da matriz de co-ocorrência conterá a quantidade de vezes que houve ocorrência do nível 0 com o nível 0 na horizontal, ou seja tanto à esquerda como a direita. No exemplo já referido o valor da posição (0,0) será 8, como se pode comprovar nas situações a seguir:

$$\begin{aligned}
 I(0,0) &\rightarrow I(0,1)I(2,3) \rightarrow I(2,4) \\
 I(0,1) &\rightarrow I(0,0)I(2,4) \rightarrow I(2,3) \\
 I(0,1) &\rightarrow I(0,2)I(4,0) \rightarrow I(4,1) \\
 I(0,2) &\rightarrow I(0,1)I(4,1) \rightarrow I(4,0)
 \end{aligned}$$

Para finalizar, para a imagem da correspondente matriz 2.1, as matrizes de co-ocorrência resultantes para $\alpha=0^\circ$, $\alpha=90^\circ$, $\alpha=45^\circ$, $\alpha=135^\circ$ seriam respectivamente as tabelas 2.3, 2.4, 2.5 e 2.6

	0	1	2
0	8	8	2
1	8	6	2
2	2	2	2

Tabela 2.3: *Matriz de co-ocorrência para $\alpha=0^\circ$*

	0	1	2
0	6	5	2
1	5	4	4
2	2	4	0

Tabela 2.4: *Matriz de co-ocorrência para $\alpha=90^\circ$*

Para se poder extrair os descritores através destas matrizes é necessário normaliza-las, sendo esse processo feito através da divisão de cada elemento da matriz original pela soma dos seus componentes, como podemos verificar pela equação a seguir:

$$P(m, n) = P(m, n) / \text{sum}[i = 0, Hg](\text{sum}[j = 0, Hg](P(i, j))) \quad (2.2)$$

	0	1	2
0	4	9	2
1	9	2	5
2	2	5	0

Tabela 2.5: Matriz de co-ocorrência para $\theta = 45^\circ$

	0	1	2
0	8	4	1
1	4	6	4
2	1	4	0

Tabela 2.6: Matriz de co-ocorrência para $\theta = 90^\circ$

Onde H_g representa o maior nível de cinza presente na imagem.

Como já foi dito a matriz de co-ocorrência é o ponto de partida para a extração de diversos descritores estatísticos, de seguida apresentamos alguns desses descritores que foram originalmente propostos por Haralick, Shanmugan e Dinstein (1973), bem como as respectivas fórmulas que as descrevem [6]:

- O *Segundo momento angular* (SMA) e Energia denotam a homogeneidade da imagem. Valores elevados de SMA ou Energia ocorrem quando a imagem está muito ordenada;

$$Energia = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} [P(i, j, \delta, \theta)]^2 \quad (2.3)$$

- Entropia, que representa a desordem em uma imagem, ou seja, o grau de desorganização de níveis de cinza;

$$Entropia = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} P(i, j, d, \theta) \log_2 [P(i, j, \delta, \theta)] \quad (2.4)$$

- Contraste ou variância, calcula a presença de mudança súbita de níveis de cinza na imagem. Valores na diagonal da matriz representam

pouco contraste, aumentado proporcionalmente com a distância da diagonal;

$$\text{Contraste} = - \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (i-j)^2 P(i, j, \delta, \theta) \quad (2.5)$$

- Correlação, que serve para medir a dependência linear de pixéis em relação a sua vizinhança;

$$\text{Correlacao} = \frac{\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} ijP(i, j, d, \theta) - \mu_x \mu_y}{\sigma_x \sigma_y}, \text{ sendo} \quad (2.6)$$

$$\mu_x = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} iP(i, j, \delta, \theta) \quad (2.7)$$

$$\mu_y = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} jP(i, j, \delta, \theta) \quad (2.8)$$

$$\sigma_x = \sqrt{\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} i^2 P(i, j, \delta, \theta) - \mu_i^2} \quad (2.9)$$

$$\sigma_y = \sqrt{\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} j^2 P(i, j, \delta, \theta) - \mu_j^2} \quad (2.10)$$

- Homogeneidade, em que os valores dos seus pesos são inversamente proporcionais aos pesos do contraste, com este a diminuir exponencialmente quanto mais distantes da diagonal. Grandes valores de homogeneidade significam que a imagem contém níveis de cinza similares.

$$\text{Homogeneidade} = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \frac{1}{1 + (i+j)^2} P(i, j, d, \theta) \quad (2.11)$$

Abordagem espectral

A abordagem espectral é principalmente utilizada para identificar a periodicidade global duma imagem através da identificação de picos de alta energia no espectro e baseia-se nas propriedades do espectro de Fourier [10]. De seguida apresenta-se a definição de transformada de Fourier bem como do seu espectro e posteriormente de uma outra técnica utilizada para a extracção de características nesta abordagem - Filtros de Gabor.

Transformada de Fourier e espectro

A transformada de Fourier é utilizada em bastantes áreas da ciência e tecnologia nos dias de hoje. Esta transformada é uma das opções para realizar uma alteração de domínio espacial em espectral, onde um sinal é caracterizado pelas suas características de frequência. Ela facilita a análise da imagem, pois mostra-a decomposta nos seus componentes de frequência, o que permite identificar os aspectos mais comuns da imagem, como as texturas.

Existem vários tipos de transformada de Fourier, mas a mais relevante para este projecto é a Transformada Discreta de Fourier (*Discrete Fourier Transform (DFT)*), que é utilizada no estudo do espectro de sinais e denota-se $F(u, v)$ que é dada pela equação seguinte:

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(\frac{ux}{M} + \frac{vy}{N})} \quad (2.12)$$

$$(u = 0, 1, \dots, M - 1, v = 0, 1, \dots, N - 1)$$

Existe um algoritmo que utiliza uma técnica que permite avaliar a DFT e a sua inversa de uma maneira mais rápida e eficaz, que contribui para um menor esforço computacional, que se chama de Transformada rápida de Fourier (*Fast Fourier Transform (FFT)*). Este algoritmo baseia-se no chamado de método de desdobramentos sucessivos, onde a Transformada de Fourier pode ser obtida por:

$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) W_N^{ux} \quad (2.13)$$

Posto isto, o principal objectivo da Transformada de Fourier será calcular o espectro. O espectro é bastante útil por possuir três características:

1. Picos salientes no espectro fornecem a direcção dos padrões de texturas;
2. O período espacial fundamental dos padrões é fornecido pela posição dos picos no plano de frequência;
3. Quando os elementos periódicos são eliminados através do processo de filtragem deixa os elementos não periódicos na imagem, que podem ser descritos por técnicas estatísticas.

O espectro é geralmente adaptado para descrever a direcção de padrões periódicos em uma imagem.

Um processo de simplificação na interpretação e detecção das características do espectro já referidas, é feito para expressar o espectro em coordenadas polares, obtendo uma função $S(r, \theta)$, sendo S uma função de espectro, r e θ variáveis no sistema de coordenadas. Uma descrição global para $S(r)$ e $S(\theta)$ é obtido através das seguintes funções:

$$S(r) = \sum_{\theta=0}^{\Pi} S_{\theta}(r) \quad (2.14)$$

$$S(\theta) = \sum_{r=1}^R S_r(\theta) \quad (2.15)$$

Em que R é o raio de um circunferência centrada na origem

Os resultados destas duas equações representam um par de valores para cada par de coordenadas (r, θ) , e com a alteração das mesmas é possível gerar funções unidimensionais $S(r)$ e $S(\theta)$ que representam descrições

de energias espectral da textura para uma imagem ou região. Os descritores das próprias funções podem ser determinados para uma caracterização quantitativa.

A figura 2.2 ilustra o uso das equações 2.14 e 2.15 para a descrição global de textura. A figura 2.2(a) mostra uma imagem com uma textura periódica, enquanto a figura 2.2(b) apresenta o espectro correspondente.

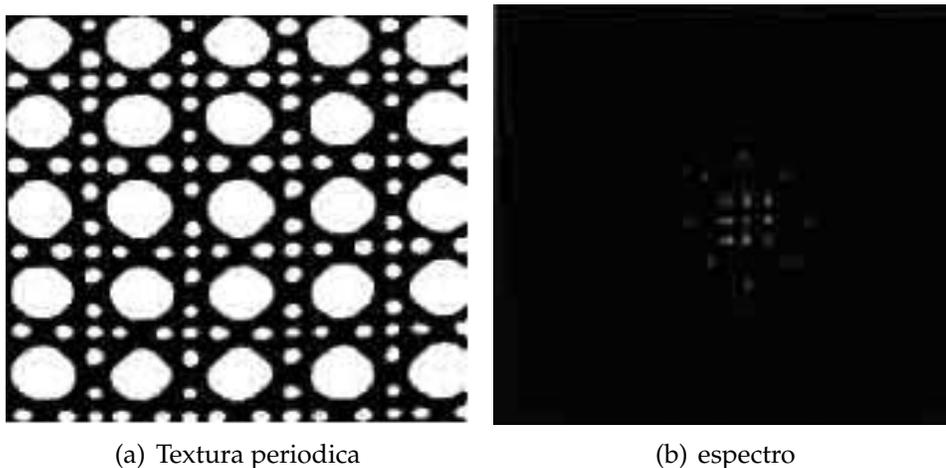


Figura 2.2: *Imagem com uma textura periódica e o respectivo espectro*

Filtros de Gabor

Outra técnica eficiente para a extracção de características na abordagem espectral é os filtros de Gabor.

Foi desenvolvido por Dennis Gabor em 1946, e consiste numa variante local da transformada de Fourier.

Os filtros de Gabor são bastante eficientes no processo de análise de textura a partir de frequências espaciais, simulando algumas das características do sistema visual humano [5] [4], tudo porque estes são capazes de caracterizar um sinal simultaneamente nos domínios temporal e das frequências, que são limitados pela relação de incerteza conjunta, ou princípio de Heizenberg:

$$(\Delta t)(\Delta \omega) \geq 1/4\pi \quad (2.16)$$

Onde, Δt $\Delta \omega$ representam a incerteza nos domínios temporal e das frequências respectivamente. Os conjuntos de funções que atingem este limite inferior de incerteza foram determinados por Gabor como sendo:

$$f(t) = \exp \left[\frac{-1}{2} \left(\frac{t}{\sigma} \right)^2 + i\omega t \right] \quad (2.17)$$

Resumidamente, a função estabelecida por Gabor descreve uma onda senoidal com frequência ω modulado por envelope gaussiano com duração σ . Gabor propôs um conjunto original de filtros que foi estendido para o caso bidimensional por Daugman [4], podendo assim ser aplicados para dados do tipo imagem, sendo esse filtro representado por:

$$f(x, y, u_0, v_0, \sigma_x, \sigma_y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp \left\{ \frac{-1}{2} \left[(x/x_0/\sigma_x)^2 + \left(\frac{y}{\sigma_y} \right)^2 \right] \right\} \exp \{2\pi i (u_0 x + v_0 y)\} \quad (2.18)$$

2.2 Análise dos componentes principais

Como foi dito anteriormente, normalmente tende-se a seleccionar os melhores atributos que melhor descrevem as propriedades dos componentes da imagem, isto é necessário porque um dos grandes problemas inerentes ao processo de identificação de regiões em imagens é a grande dimensionalidade do vector de características, pois compromete o desempenho e a exactidão do classificador [8]. A redução da dimensionalidade consiste em diminuir o conjunto de características, sendo esta possível porque algumas das características podem não ser relevantes para a tomada de decisão, e assim é possível identificar o número mínimo de características que é suficiente para descrever uma região ou identificar um objecto na imagem.

Existem duas abordagens para se reduzir a dimensionalidade que são a extracção e a selecção de características. A primeira cria novas características a partir de combinações ou transformações do conjunto original, enquanto a selecção de características visa determinar um subconjunto de

características capazes de discriminar objectos através de regras previamente estabelecidas.

O método utilizado neste trabalho é a análise de componentes principais que é um método estatístico bastante utilizado na área de reconhecimento de padrões por ser simples e rápido na sua execução. A PCA pode ser entendido como um transformação linear de m características originais em m características novas e estas são ordenadas de acordo com a sua relevância.

Segundo Kendall (1980), a PCA pode ser utilizada para várias tarefas das quais destaco a simplificação do conjunto de dados, que consiste em descobrir uma forma simplificada de representar o universo de estudo, podendo ocorrer através da transformação, sendo uma combinação linear ou não, de um conjunto de variáveis autónomas em outro conjunto autónomo e de inferior dimensão [9].

De seguida, para uma melhor compreensão do funcionamento da PCA apresenta-se um exemplo adaptado de Smith (2002) [14]. Seja o espaço bidimensional composto por duas características, cada uma com dez amostras, conforme se verifica na tabela 2.7

X	2.5	0.5	2.2	1.9	3.1	2.3	2.0	1.0	1.5	1.1
Y	2.4	0.7	2.9	2.2	3.0	2.7	1.6	1.1	1.6	0.9

Tabela 2.7: Conjunto com dez amostras para as duas características

De seguida é calculada a média aritmética através da equação 2.19, obtendo-se os valores $\bar{X} = 1.81$ e $\bar{Y} = 1.91$.

$$\bar{X} = \frac{\sum_{i=1}^N X_i}{N} \quad (2.19)$$

Subtraindo de cada amostra da tabela 2.7 pelo resultado da média, obtêm-se a tabela 2.8.

A partir da tabela 2.8 constrói-se a matriz de co-variância, sendo que esta utiliza os conceitos de variância, desvio padrão e co-variância com as equações 2.20 2.21 2.22 respectivamente.

X	0.69	-1.31	0.039	0.09	1.29	0.49	0.19	-0.81	-0.31	-0.71
Y	0.49	-1.21	0.99	0.28	1.09	0.79	-0.31	-0.81	-0.31	-1.01

Tabela 2.8: subtração da média em cada amostra

$$\sigma^2 = \frac{\sum_{i=1}^N (X_i - \bar{X})^2}{N - 1} \quad (2.20)$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^N (X_i - \bar{X})^2}{N - 1}} \quad (2.21)$$

$$Cov(x, y) = \frac{\sum_{i=1}^N (X_i - \bar{X})(Y_i - \bar{Y})}{N - 1} \quad (2.22)$$

Posto isto, a matriz de co-variância com dimensão n é dada por:

$$C_{N \times N} = (C_{i,j}, C_{i,j} = cov(Dim_i, Dim_j)) \quad (2.23)$$

Para o exemplo já referido a matriz de co-variância é:

$$\begin{pmatrix} 0.616555556 & 0.615444444 \\ 0.615444444 & 0.716555556 \end{pmatrix}$$

A PCA baseia-se na transformação linear das observações originais, sendo esta conhecida na Álgebra Vectorial como generalização de autovectores e autovalores. Os autovalores são medidos em unidade de variância e caracterizam os comprimentos dos eixos das componentes principais e para cada um existe um vector de módulo unitário chamado autovector que representa a direcção dos eixos das componentes principais.

Sendo $T(v) = m.v$, diz-se que v é um autovector de T com relação ao autovalor m . Os autovalores são calculados através das equações 2.24 e 2.25.

$$|S - \lambda I| = 0 \quad (2.24)$$

Onde λ é um escalar

$$u_i = \frac{t_i}{\sqrt{t^T t_i}} \quad (2.25)$$

Para $i=1,2,\dots,p$, tal que t são os autovalores, S é a matriz de co-variância e I é a matriz identidade.

Assumindo para o exemplo $i=1$, obtém-se:

$$|S - I\lambda| = \begin{pmatrix} 0.616555556 - \lambda & 0.615444444 \\ 0.615444444 & 0.716555556 - \lambda \end{pmatrix} = 0$$

Resolvendo o determinante, obtém-se:

$$\lambda_1 = 0.049083399 \text{ e } \lambda_2 = 1.284027713$$

$$|S - I\lambda_1| = \begin{pmatrix} 0.616555556 - 0.049083399 & 0.615444444 \\ 0.615444444 & 0.716555556 - 0.049083399 \end{pmatrix} = \begin{pmatrix} t_{11} \\ t_{21} \end{pmatrix} = 0$$

Sendo $t_{11}=1$, então $t_{21} = 0.922052611$, resultando

$$u_1 = \begin{pmatrix} -0.735178655 \\ 0.677873399 \end{pmatrix}$$

De maneira idêntica, sendo $t_{21}=1$,

$$u_2 = \begin{pmatrix} -0.677874499 \\ 0.735178656 \end{pmatrix}$$

Com os autovectores, cria-se a matriz U :

$$U = \begin{pmatrix} -0.735178655 & -0.677873399 \\ 0.677873399 & -0.735178656 \end{pmatrix}$$

Para obter o novo conjunto de dados, utiliza-se a equação 2.26.

$$Z_i = [u_i^t[X - \bar{X}]] \quad (2.26)$$

Para o exemplo fica:

$$Z_1 = \begin{pmatrix} -0.842100707 \\ -0.17568636 \end{pmatrix}$$

Posto isto, calculando a fórmula para todas amostras da tabela 2.8 , o conjunto de novos componentes resultantes é dado pela tabela 2.9.

X	-0.842	1.777	-0.992	-0.274	-1.675	-0.912	0.099	1.144	0.438	1.23
Y	-0.175	0.142	0.384	0.130	-0.209	0.165	-0.349	0.464	0.017	-0.162

Tabela 2.9: conjunto de novos componentes

Para finalizar, a partir do novo conjunto de dados e dos autovectores seleccionam-se os valores mais relevantes de acordo com a ordem crescente dos autovalores.

2.3 Redes Neurais Artificiais

As Redes neuronais artificiais (*Artificial Neural Network* (ANN)) , é um modelo computacional que tenta simular a estrutura e os aspectos funcionais das redes neuronais biológicas, inclusive o seu comportamento, ou seja, aprender, errar e descobrir.

É constituído por um grupo de neurónios artificiais interligados, e a informação é processada através de uma abordagem de ligação computacional. Na maioria dos casos as redes neuronais artificiais são um sistema adaptativo que muda a sua estrutura com base em informações internas ou

externas que vão passando pela rede durante o processo de aprendizagem. Normalmente são utilizadas para modelar relações complexas entre dados de entrada e saída ou para encontrar padrões em dados.

Para representar as redes neuronais artificiais utiliza-se um grafo direcionado, onde os nodos são denominados de elementos de processamento, representando os neurónios, e as arestas representam a transmissão do sinal entre um neurónio e outro.

Uma rede neuronal pode conter uma ou várias camadas. Por exemplo, com três camadas, temos a camada de entrada, em que as unidades recebem os padrões; a camada intermédia, onde é feito o processamento e a camada de saída, que conclui e apresenta o resultado final. Quanto maior o número de camadas, melhor a capacidade de aprendizagem.

O número de camadas descreve a aptidão de representação das relações entre o espaço de entrada e o de saída. A inexistência de uma camada intermédia, característica do modelo perceptron, condiciona-o a representar bem apenas as relações linearmente independentes. A existência de camadas intermédias retira essa mesma limitação [15]. Na figura 2.3 apresenta-se um diagrama simplificado de uma rede neuronal com 3 camadas, a camada de entrada com duas unidades, a camada intermédia com 5 unidades e uma camada de saída.

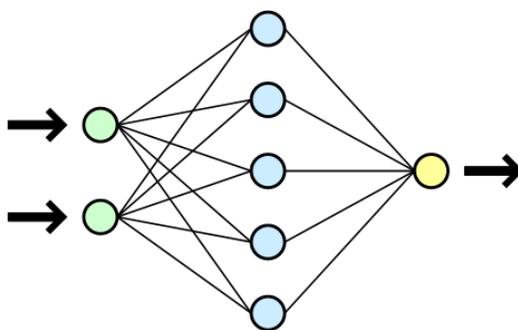


Figura 2.3: Diagrama simplificado de uma Rede neuronal

Uma rede neuronal tem de ser configurada de maneira a que quando aplicado um conjunto de dados de entrada o resultado obtido seja o espe-

rado. Existem vários métodos para aproveitar os pontos fortes das ligações existentes. Uma das maneiras é treinar a rede neuronal oferecendo-lhe padrões de ensino e deixa-la modificar os pesos de acordo com alguma regra de aprendizagem.

A este método chama-se aprendizagem supervisionada, onde são fornecidos padrões de entrada e a saída correspondente a esses padrões. A figura 2.4 apresenta um possível esquema deste método.

Backpropagation é um algoritmo que se baseia na aprendizagem supervisionada, que resumidamente começa com um padrão a ser submetido na rede e este vai avançando camada por camada até a resposta ser produzida pela camada de saída, depois esta é comparada com a saída esperada para esse padrão particular. Se estiver errado, o erro é calculado e propagado desde a camada de saída até a camada de entrada e os pesos das conexões das unidades das camadas internas vão sendo ajustadas consoante o erro é retro propagado [1].

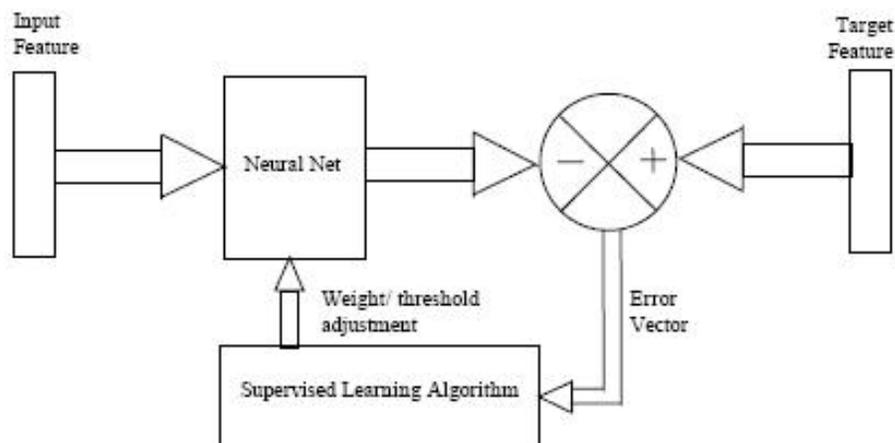


Figura 2.4: Esquema representativo do método de aprendizagem supervisionado

Capítulo 3

Trabalho desenvolvido

Como já foi dito, o desenvolvimento de um método para uma análise eficiente de imagens é uma tarefa complicada, não existindo um método que apresente um desempenho satisfatório em vários domínios de aplicação.

Neste capítulo descrevem-se as várias etapas que correspondem ao método que foi utilizado para a construção do sistema automático de análise e quantificação de regiões edificadas, com base nas respectivas características de texturas.

Essas etapas encontram-se descritas na figura 3.1, sendo elas: aquisição de dados, pré-processamento, extracção de características, normalização, análise das componentes principais, classificação e por último pós-processamento.

3.1 Aquisição de Dados

Nesta fase foi necessário desenvolver uma aplicação para extrair as imagens do *Google Maps* e para isso utilizou-se a API *static maps* do *Google Maps* [2], construindo-se uma aplicação usando a linguagem C#.

O funcionamento da aplicação é bastante simples mas exige um conhecimento á priori do funcionamento do *static maps*, que utiliza um sistema de coordenadas para definir o globo terrestre sendo que estas coordenadas

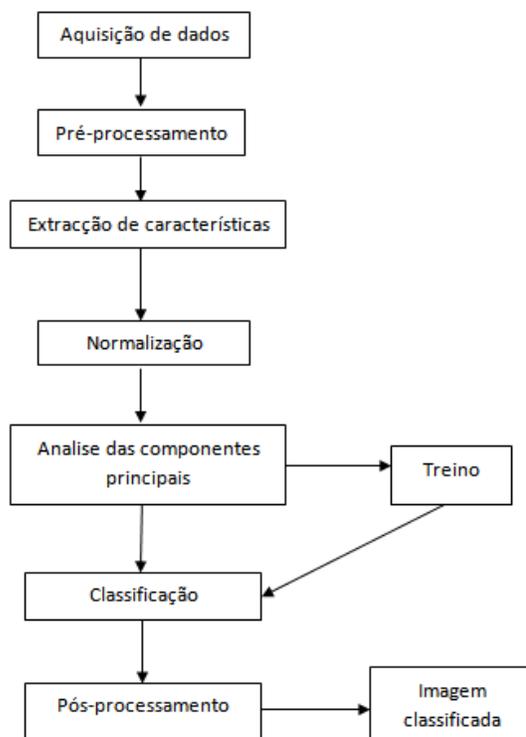


Figura 3.1: Esquema das várias etapas do método proposto

vão variando consoante o nível de zoom a que se encontra a imagem que se está a visualizar. Posto isto, na aplicação é necessário ter em conta estes três parâmetros quando inserido pelo utilizador, pois as imagens resultantes dependem da distância dos dois pontos, sendo que a resolução da imagem será igual á distância entre os mesmos e para isso é necessário que os pontos inseridos estejam de acordo com a altitude inserida.

A figura 3.2 mostra a interface da aplicação onde se pode verificar que o utilizador terá de inserir dois pontos em coordenadas *Global Positioning System* (GPS) (latitude, longitude, latitude2 e longitude2), bem como o nível de altitude correspondente a essas coordenadas, e ao clicar em obter tiles irá obter as imagens referentes daquele ponto na pasta inserida em localização de destino.

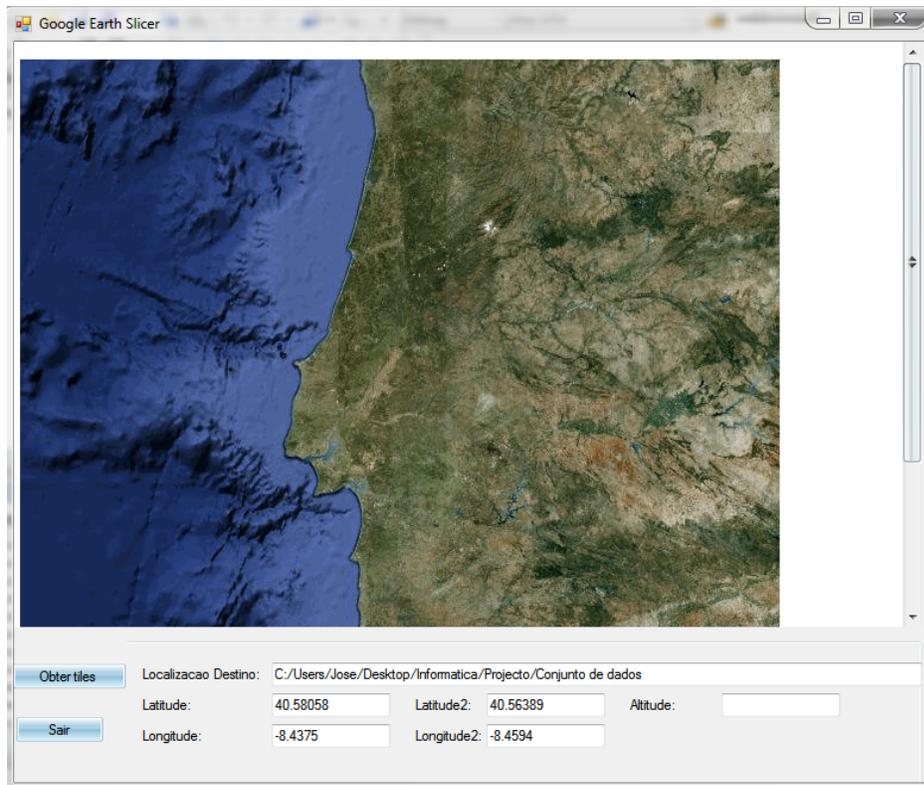


Figura 3.2: Interface da aplicação Google Earth Slicer

Das imagens extraídas foram seleccionadas algumas imagens de zonas diferentes, mais propriamente de duas zonas distintas, uma da zona mais rural 3.3(b), onde o contraste entre os edifícios e a natureza é mais acentuado, e outras de zonas urbanas 3.3(a) onde existem mais edifícios. Na figura 3.3 encontra-se dois exemplos de duas imagens utilizadas no projecto. No capítulo resultados é possível verificar a diferença que se obteve em termos de resultados finais entre estes dois tipos de imagens, como seria de esperar, uma vez que nas imagens do meio urbano o contraste entre os edifícios com o resto não é tão acentuado como no meio rural.

Antes de avançar para o processamento das imagens foi necessário decidir as classes, isto é, numa imagem o que representa aquilo que queremos classificar e identificar. Por isso optou-se por definir como tendo valor 0 toda a região que não se pretende identificar, e com valor 1 como sendo



(a) Zona urbana

(b) Zona rural

Figura 3.3: *Exemplo dos dois tipos de imagens utilizados*

regiões que se quer identificar (casas, edifícios, construções humanas). Depois de definidas as classes para cada imagem existente num conjunto de treino, ou seja, as imagens que vão servir para treinar o classificador como veremos mais à frente, foi necessário proceder à construção de uma imagem onde as regiões que não interessam identificar fossem preenchidas a preto e as regiões com edifícios preenchida a branco. Para isso utilizou-se o software Photoshop para ajudar no processo e "pintura" das imagens binárias onde a 1 temos as regiões edificadas e a 0 todo o resto. A estas imagens passaremos a chamar de máscara. Na figura 3.4 encontra um exemplo de uma imagem 3.4(a) e a respectiva máscara 3.4(b).

3.2 Pre-Processamento

Nesta etapa as imagens são preparadas para as próximas fases, para permitir que a imagem que vai ser processada se encontre mais apropriada do que a imagem original.

Nesta fase do projecto foram realizadas duas operações, a conversão das imagens originais para níveis de cinza e HSV.

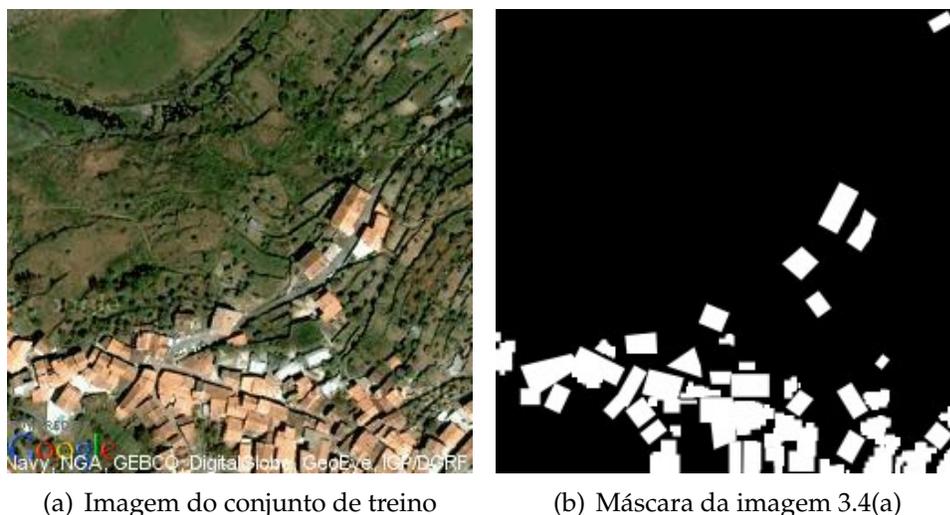


Figura 3.4: Imagem e respectiva máscara

As imagens presentes na base de dados encontram-se no sistema de cores *Red, Green, Blue* (RGB), e é necessário convertê-las para o sistema de cores níveis de cinza para nas etapas seguintes se processar as imagens apenas neste sistema de cores, pois este representa a imagem de uma forma simplificada onde o valor de cada pixel é um exemplo único, isto é, apenas é carregada a informação de intensidade tornando assim o processamento dos vários algoritmos existentes nas fases seguintes muito mais rápidos e eficientes. Esta conversão é também necessária, pois o valor dos pixels da imagem neste sistema irá representar uma das características existentes.

A conversão da imagem original no sistema de cores RGB para HSV foi necessária pois as componentes da cor de um objecto existentes na imagem original são correlacionadas com a quantidade de luz que atinge o objecto e por isso uns com os outros, o que proporciona aos descritores da imagem uma definição difícil do objecto. Por isso o valor dos três componentes existentes no sistema de cor HSV que são consideradas semelhantes na abordagem e intenção com o processamento neuronal utilizados pela visão de cores do ser humano, são extremamente eficazes quando o objectivo é detectar objectos. Assim sendo estas três componentes foram utilizadas como características no presente projecto [16].

Na figura 3.5 encontra-se uma imagem nos diferentes sistemas de cores.

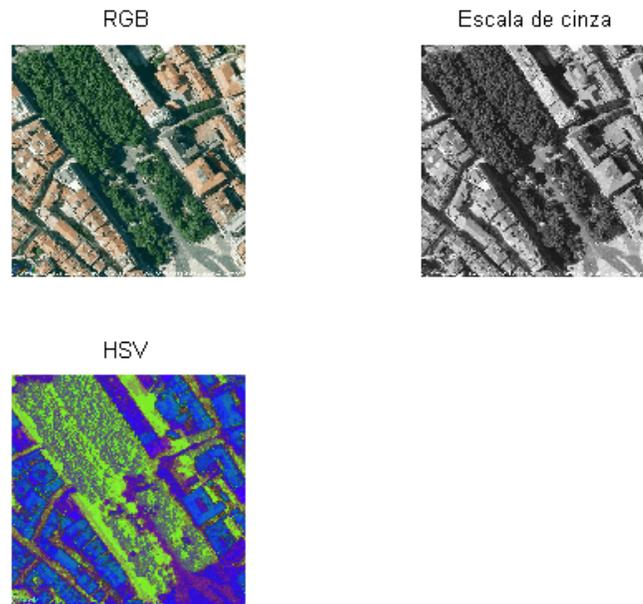


Figura 3.5: *Diferentes sistemas de cores*

Para efectuar as conversões entre os diferentes sistemas de cor recorreu-se a duas funções existentes na toolbox do Matlab que são: *rgb2gray* e *rgb2hsv*.

Para finalizar nesta fase foi ainda necessário efectuar a conversão entre tipos, sendo alterado do tipo *uint8* para *double*, pois alguns dos algoritmos utilizados nas fases seguintes requerem os dados como tipo *double*. Mais uma vez recorreu-se a uma função existente na toolbox do matlab, sendo esta: *im2double*.

3.3 Extração de características

Esta fase representa uma das mais importantes, da abordagem utilizada para o desenvolvimento do sistema capaz de identificar regiões edificadas numa imagem, pois é aqui que as características presentes na imagem vão ser extraídas para nas fases seguintes ser possível classificar e obter os resultados esperados.

Como já foi expresso no capítulo anterior as características extraídas neste projecto são obtidas através dos momentos de Hu e dos descritores de texturas, sendo que dentro destes temos as características retiradas da matriz de co-ocorrência, bem como dos filtros de Gabor e do espectro de Fourier. Todas estas características à medida que se calculam adicionam-se ao vector de características sendo este de dimensão $l*c*n$ onde "l" e "c" representam o tamanho da imagem original e "n" o número de características extraídas.

De seguida irei explicar como cada uma destas características foram obtidas, mas antes explica-se o processo de percorrer uma imagem com um filtro que é utilizado ao longo das várias etapas da extração de características.

Filtro e moldura

Sempre que se calculou características presentes numa imagem foi necessário percorre-la com um filtro, começando num ponto da imagem até chegar ao fim desta. O método utilizado para realizar este processo consiste em criar um filtro de tamanho "n" que irá se posicionar no primeiro quadrante de tamanho "n" da imagem e ir avançando pixel a pixel, percorrendo em primeiro as colunas e depois as linhas, até chegar ao fim da imagem e em cada iteração consoante a característica que tiver a ser calculada esta é obtida e colocada na posição do pixel do meio do filtro, ou seja, em cada posicionamento do filtro são calculadas as características referentes ao pixel que se encontra no meio com os seus pixels vizinhos que compõem o filtro. Um exemplo do funcionamento deste processo encontra-se na figura 3.6. Onde podemos ver em 3.6(a) que representa a 1ª iteração do filtro com $n=3$, onde este está colocado no início da imagem e o valor calculado irá

ser colocado numa nova matriz no lugar do pixel do meio representado pelo valor 1 na imagem, em 3.6(b) verificamos a 2ª iteração depois de haver uma avanço do filtro em relação às colunas e em 3.6(c) verifica-se a que o filtro está colocado depois de ter percorrido todas as colunas e avançou uma linha.

Para a realização deste processo é necessário criar uma moldura à imagem para que quando se coloca o filtro no início da imagem ou no fim, o filtro se encontre posicionado para calcular as características referentes aos pixéis da primeira linha e coluna pois sem isso os primeiros pixéis seriam ignorados pois em cada iteração a valor calculado diz respeito à posição do centro do filtro. Para criar a moldura optou-se por duplicar as primeiras linhas e colunas da imagem consoante o filtro, isto é, para um filtro de tamanho $n \times n$, e tendo em conta que este filtro será sempre ímpar, o número de linhas e colunas a serem duplicadas será igual a $(n-1)/2$. Para o exemplo da figura 3.6 temos um filtro com $n=3$ logo o número de colunas e linhas a duplicar da imagem original seria igual a 1 ($(3-1)/2$).

O algoritmo 1 representa uma possível implementação deste processo de percorrer a imagem com um filtro.

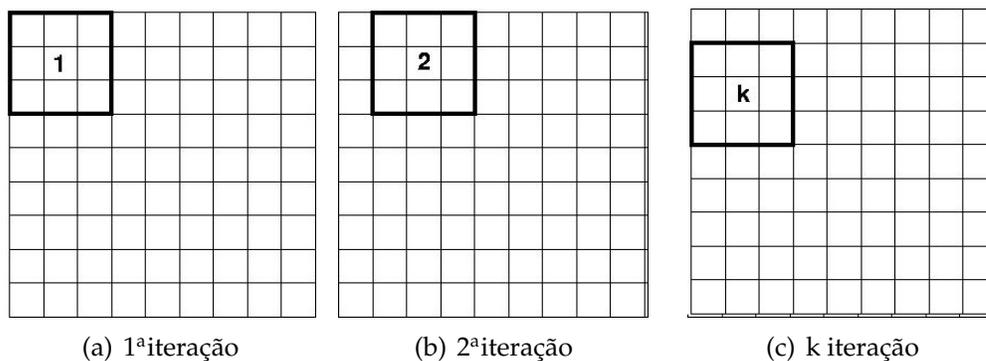


Figura 3.6: Exemplo do processo de percorrer uma imagem

Momentos de Hu

Para o cálculo dos momentos de Hu utilizou-se uma função, que dada uma imagem devolve os sete momentos de Hu utilizando as funções já

Algoritmo 1 Este algoritmo é responsável por percorrer uma imagem e em cada iteração criar um filtro de tamanho n

n = tamanho do filtro

IM= imagem original com moldura de acordo com $(n-1)/2$

```
for  $i = 1$  to  $l$  do
  for  $j = 1$  to  $c$  do
    filtro=IM( $i$  até  $i+n-1$  ;  $j$  até  $j+n-1$ )

    calcular características

     $i=i+1$ 

     $j=j+1$ 

  end for
end for
```

apresentadas neste relatório no capítulo 2. Através do algoritmo 1 utilizou-se um filtro de tamanho 5 para percorrer a imagem e para cada iteração os sete momentos de Hu são calculados, através da função já referida, para a imagem referente ao filtro e sendo o resultado guardado numa nova matriz na posição relativa ao pixel do centro, obtendo como resultado final uma matriz com o tamanho $l*c*7$, onde " l " e " c " são o número das linhas e colunas da imagem original respectivamente.

Matriz de co-ocorrência

Mais uma vez algoritmo 1 foi utilizado para percorrer a imagem com um filtro e para cada iteração extrair as características existentes na matriz respeitante ao filtro é utilizado, sendo que desta vez o filtro tem um tamanho de 15 por 15 visto que aqui interessa ter um filtro maior pois para cada um desses filtros a matriz de co-ocorrência irá ser calculada. A matriz de co-ocorrência é então calculada para as quatro direcções possíveis (0° , 45° , 90° e 135°) considerando um vizinho, o que na prática resulta em quatro matrizes, como já foi explicado no capítulo 2. Depois de obter as quatro

matrizes, a média destas é calculada e é então obtida uma única matriz de co-ocorrência que vai servir para calcular as características referentes a esta.

O passo seguinte é então extrair os diversos descritores estatísticos, a partir da matriz co-ocorrência calculada. São calculados ao todo 22 descritores, sendo que os mais relevantes se encontram descritos na secção da matriz de co-ocorrência do capítulo 2. Por fim e ainda dentro da iteração que percorre a imagem, estes 22 descritores são colocados na posição relativa ao pixel que se está a tratar, em 22 novas matrizes, que no fim da imagem toda percorrida, dizem respeito a uma matriz com o tamanho $l*c*22$, ou seja, 22 matrizes do tamanho da matriz correspondente à imagem original mas cada uma respeitante aos 22 descritores calculados a partir da matriz de co-ocorrência.

Filtros de Gabor

Os filtros de Gabor, como vimos no capítulo 2, são bastante úteis para a extracção de características de texturas, por isso a partir desta técnica foram obtidas 30 características.

A partir de uma função que implementa o filtro desenvolvido por Daugman [4], que se encontra demonstrado no capítulo 2 deste presente documento, as características são calculadas fazendo variar alguns dos parâmetros existentes no filtro, nomeadamente, a frequência da função senoidal bem como a orientação do filtro de gabor. Para a frequência os valores assumidos são 0,2,4,8,16 e 32, e para a direcção temos $\theta = 0$, $\theta = \pi/3$, $\theta = \pi/6$, $\theta = \pi/2$ e $\theta = 3 * \pi/4$.

Para cada direcção calcula-se as respectivas imagens depois de aplicado o filtro de Gabor para todas as frequências, obtendo assim para cada direcção 6 imagens sendo uma para cada frequência. Depois para cada uma das imagens obtidas utiliza-se o algoritmo 1 para a percorrer estas com um filtro de tamanho 5 e em cada iteração do filtro a média desse é calculada e o valor obtido representa a características extraídas através dos filtros de Gabor.

Resumindo no fim de todas as imagens serem calculadas e para cada

uma destas a média também calculada através do filtro que percorre a imagem, obtemos 30 matrizes todas do tamanho da matriz correspondente à imagem original que correspondem a 30 características extraídas através dos filtros de Gabor que posteriormente se acrescenta ao vector de características.

Espectro de Fourier

Para calcular o espectro de Fourier primeiro tem de se calcular a sua transformada, para isso utilizou-se mais uma vez uma função que implementa a FFT e que existe na toolbox do matlab, sendo a sintaxe desta $F = \text{fft2}(f)$, em que "f" representa a matriz com M*N e "F" representa a transformada dessa imagem também de tamanho M*N. O passo seguinte consiste em mover a origem da transformada para o centro do rectângulo de frequência. Para isso usa-se a função com a seguinte sintaxe $F_c = \text{fftshift}(F)$, em que "F" é a transformada e "F_c" representa a transformada centralizada. Esta função opera trocando os quadrantes de "F". Posto isto, já podemos obter o espectro de Fourier através da função abs, com a sintaxe $S = \text{abs}(F_c)$, onde "S" representa o espectro de Fourier.

O próximo passo consiste em aplicar o algoritmo 1 novamente para a matriz correspondente ao espectro de Fourier. O filtro que percorre essa matriz é de tamanho 15 por 15, sendo que em cada iteração este mesmo filtro irá ser percorrido por um outro filtro de tamanho 5 por 5 onde para cada uma das iterações é calculada a média e desvio padrão correspondentes aos valores existente no primeiro filtro, ou seja, o algoritmo 1 é aqui aplicado duas vezes, a primeira para o espectro e a segunda para o filtro que percorre o espectro. O resultado da média e desvio padrão é então colocado numa matriz que em cada iteração irá guardar os 18 valores extraídos, correspondentes a 9 médias e 9 desvios padrões, o que resulta em mais 18 características colocadas no vector de características.

3.4 Normalização

Nesta fase o vector de características encontra-se preenchido com um total de 81 características, mas estas necessitam de ser normalizadas pois os intervalos entre as diversas características é diferente e como umas não devem predominar sobre as outras devido a essa variabilidade relativa a fase de normalização torna-se então bastante importante. Para isso mais uma vez utilizou-se uma função existente na toolbox do matlab sendo esta *mapminmax*, que normaliza o conjunto de dados para o intervalo [0,1] tendo em conta o máximo e o mínimo de cada coluna.

Outra operação foi efectuada durante esta fase, designadamente a existência de *Not a Number* (NAN) e a consequente substituição destes pela mediana das linhas onde o NAN se encontra.

3.5 Analise das componentes principais

Depois de ter o vector de características preenchido, a fase seguinte é seleccionar as melhores características, tendo em conta as que melhor descrevem as propriedades dos componentes da imagem, para isso utilizo um método que permite fazer uma selecção destas mesmas características, chamado PCA, que se encontra descrito no capítulo 2 deste documento.

Esta fase é, também, bastante importante pois ajuda no desempenho do classificador tornando-o mais eficaz, pois algumas das características podem não ser relevantes para a tomada de decisão e ainda torna o custo computacional mais baixo pois como esta reduz o tamanho do vector de características faz com que as próximas fases tenham de lidar com muito menos dados.

Para a implementação do PCA utilizou-se a função existente na toolbox do matlab com a sintaxe $[coef, score, variance] = princomp(T)$, onde "T" é uma matriz de tamanho N*P, onde as linhas de "T" são observações da imagem e as colunas as variáveis que correspondem às características. Variance contém os eigenvalues por ordem de crescente desde a componente mais importante para a menos importante e coef contém os coeficientes para

cada componente principal. Através destes resultados é possível obter as características mais relevantes, para isso são usadas duas abordagens, uma consiste em obter as percentagens dos valores existentes em variance e a partir de uma certa percentagem obter k componentes principais; ou então opta-se por definir logo um " k " que representa o número de componentes principais que se pretende. Por exemplo a primeira abordagem consiste em definir 97% dos componentes principais, e o que é feito é somar as percentagens respeitantes os valores existentes em variance desde o primeiro valor até se atingir os 97%, quando esta percentagem é atingida verifica-se o número de componentes já calculadas que será então o " k " e perante esse valor selecciona-se as primeiras " k " ocorrências existentes em *coef*. Para o segundo método define-se logo um " k ", por exemplo 10 e selecciona-se as primeiras 10 ocorrências de *coef*. Para estes dois métodos vamos obter resultados diferentes, sendo que estes são expostos no capítulo resultados.

Por último a partir do resultado obtido depois de escolher o número de componentes que se pretende, resta multiplicar cada linha da matriz que contem todas as características e todas as observações da imagem, ou seja cada instância do conjunto de características extraído da imagem pelo resultado obtido do PCA.

3.6 Treino e classificação

Depois de seleccionadas as características principais a fase que se segue é a de classificação, mas antes foi necessário acrescentar à matriz com as características principais para cada instância da imagem, o respectivo valor da classe a que cada instancia pertence, isto é, para cada instância atribuiu-se conforme o valor (0 ou 1) presente na respectiva mascara da imagem do conjunto de treino.

O classificador utilizado foi redes neuronais, que se encontra descrito no capítulo 2 deste documento. Este classificador é composto por várias etapas. Começa-se por separar em função das duas classes existentes (0 ou 1) a matriz que contem os dados relativos à selecção que foi feita na fase anterior, ou seja, a partir da matriz devolvida pela fase de selecção das características principais e depois de a cada instancia ter sido atribuído

a classe correspondente de acordo com a máscara, obtemos assim duas matrizes, uma com as instâncias correspondentes a 0, ou seja que não representa edifícios e outra com as restantes representando edifícios. O passo seguinte é seleccionar o conjunto de treino e teste da imagem, isto é, da imagem que se está a processar selecciona-se instâncias umas para treino e outras para teste, sendo que nos testes realizados foram divididos 5000 instâncias para treino e as restantes para teste. Dentro das 5000 instancias 60% destas são de valor 0 e 40% de valor 1, ou seja, 3000 instancias que não representam a região que se pretende identificar e 2000 dessa região. De salientar que antes de se escolher estas instâncias o conjunto de dados foi baralhado para assim evitar repetições e tornar as instancias escolhida as mais amplas possíveis. Posto isto, obtemos um conjunto de treino com 5000 instâncias e um conjunto de teste com as restantes instâncias da imagem.

O próximo passo consiste em inicializar a rede através da função *newff*. Esta função cria uma nova rede *feedforward* com *backpropagation*. Como parâmetros recebe os mínimos e máximos de cada linha do conjunto de treino que se vai utilizar para treinar a rede. O próximo parâmetro é um vector com tantos elementos quanto camadas, sendo que se optou por usar três camadas: a de entrada, intermédia e saída, o número de cada uma delas diz respeito ao número de neurónios que se pretende. No trabalho efectuado o número de neurónios da camada de entrada é o número de características seleccionadas na fase anterior, na camada de saída existem duas opções que representam as duas classes existentes, ou seja, uma representa a região que se pretende identificar e a outra o contrário, quanto a camada intermédia o número de neurónios vai-se modificando e testando os resultados obtidos, para se prever com que número se obtêm o melhor resultado. No capítulo resultados, demonstram-se os vários resultados obtidos com a variação deste parâmetro, sendo que o intervalo variado foi de $k/2$ até $k*2$, com "k" igual ao número de neurónios da camada de entrada e conseqüente o número de características utilizadas para descrever a imagem. O próximo parâmetro da função *newff*, indica as funções de activação utilizadas para cada uma das camadas existentes no parâmetro anterior. Para a camada de entrada e intermédias utilizou-se a função linear (*purelin*), e para a camada de saída utilizou-se a função tangente hiperbólica (*tansig*). O último parâmetro diz respeito ao algoritmo de treino sendo este o *traingdx* - Gradient descent backpropagation com momentum e taxa

adaptativa.

Depois da rede criada, é necessário ajustar alguns parâmetros referentes a esta, para que depois seja feito o treino, sendo esses, o número de epochs, a taxa de aprendizagem, e o objectivo.

Posto isto, o próximo passo é fazer o treino que acontece invocando a execução da função *train*, que recebe como parâmetros a rede criada, bem como o conjunto de treino e ainda o conjunto que contem classes para cada instância do conjunto de treino. Na figura 3.7 encontra-se um processo de treino de uma rede neuronal utilizado no trabalho.

Por fim utiliza-se a função *sim* para simular através da rede criada um possível resultado para um conjunto de dados que se pretende classificar. Nesta função utiliza-se o conjunto de teste que corresponde ao resto da imagem que não foi seleccionado para conjunto de treino. Depois desse conjunto de teste ser classificado através da rede que foi previamente treinada, resta comparar o resultado obtido com o esperado para ver a taxa de sucesso e de erro, bem como alguns valores estatísticos, como os elementos constituintes da matriz confusão e as taxas necessárias para calcular a curva ROC. Os resultados obtidos são discutidos e apresentados no capítulo 4 deste documento.

A função *sim* também é utilizada para produzir resultados a partir de imagens do conjunto de teste, das quais a partir do resultado obtido da simulação utilizando uma rede treinada com uma das imagens do conjunto de treino, obtêm-se uma imagem binária com as regiões pretendidas identificar a branco e o resto a preto. Mais uma vez no capítulo 4 encontram-se alguns resultados obtidos.

3.7 Pós-processamento

Esta fase diz respeito ao tratamento das imagens devolvidas pelo processo de classificação. O que é feito é, através de operações morfológicas, tentar remover áreas da imagem que não lhe deviam pertencer, isto porque normalmente uma imagem classificada contém pontos isolados mal classi-

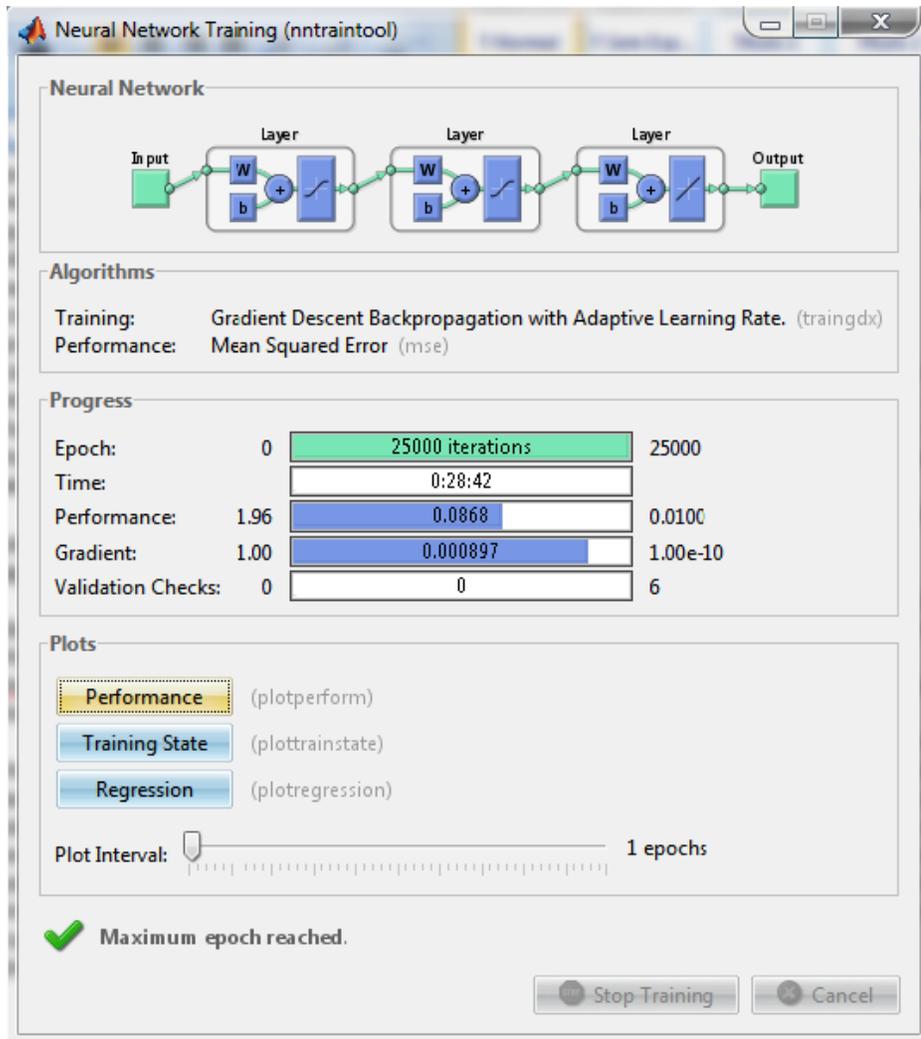


Figura 3.7: Exemplo de um treino em uma rede neuronal

ficados ou então buracos dentro de uma região bem identificada ou ainda objectos algo deformados e por isso utiliza-se operações que irão remover ruídos e pontos isolados, suavizar a imagem e recuperar a forma dos objectos. As operações em causa são a dilatação e a erosão, que combinadas dão origem a duas novas operações, sendo elas: fecho, que corresponde a uma dilatação e de seguida uma erosão, e com esta operação é possível suavizar contornos e eliminar pequenos buracos; abertura, corresponde a

uma erosão seguida de uma dilatação, proporciona remoção de ruídos e a recuperação da forma dos objectos.

Em matlab todas estas operações estão disponíveis pela *toolbox*, e para se executarem é necessário definir um elemento estruturante.

Capítulo 4

Resultados

4.1 Resultados para o conjunto de treino

Nesta secção demonstra-se os resultados obtidos a partir do conjunto de treino, sendo que para a realização de testes procedeu-se a alteração de dois parâmetros:

1. **Número de características seleccionadas no PCA;** este parâmetro varia conforme as duas abordagens já explicadas no capítulo 3 deste documento, a primeira em que selecciona-se 97% das características correspondendo em média a quatro características e a outra em que o número de características seleccionadas varia entre o intervalo de 10 até 50. Este parâmetro foi considerado pois a selecção de 97% das melhores características revelou-se não ser o melhor resultado pois poucas características são seleccionadas, como se pode verificar nos resultados a seguir apresentados e por isso testou-se com mais características seleccionadas onde se obteve melhores resultados;
2. **Número de neurónios na camada intermédia da rede neuronal;** depois de seleccionado o melhor número de características faz-se variar o número de neurónios entre o intervalo de $k/2$ e $k*2$ com k igual ao número de neurónios da camada de entrada;

Para demonstrar os resultados obtidos seleccionou-se quatro imagens do conjunto de treino, sendo estas as imagens 4.1(a),4.1(b),4.1(c) e 4.1(d).



(a) Imagem 1



(b) Imagem 2



(c) Imagem 3



(d) Imagem 4

Figura 4.1: 4 imagens do conjunto de treino

Na tabela 4.1 apresenta-se a taxa de erro para todos os testes efectuados variando o número de características seleccionadas durante a fase do PCA, de seguida compara-se os resultados entre as várias imagens e posteriormente analisa-se uma a uma.

Imagens	Número de características seleccionadas no PCA					
	97%	10	20	30	40	50
1	18.089	16.976	15.222	12.756	11.618	12.423
2	17.910	12.922	12.830	11.449	11.816	12.312
3	10.453	9.678	9.321	9.229	8.1852	8.834
4	30.084	20.499	19.619	15.675	15.062	16.143

Tabela 4.1: Resultados das taxas de erro consoante as características seleccionadas pelo PCA

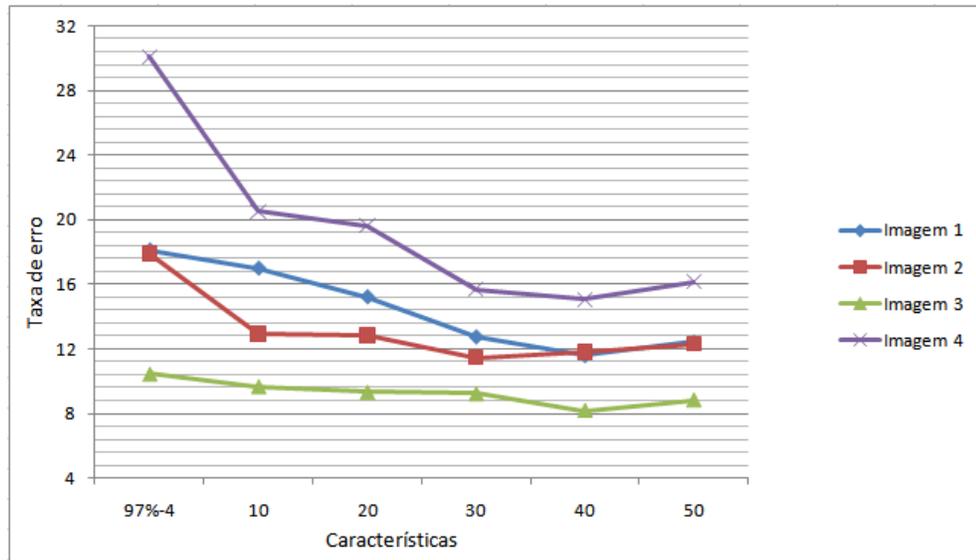


Figura 4.2: Gráfico das taxas de erro para várias características

A imagem 4.1(c) foi aquela que obteve um melhor resultado, ou seja uma taxa de erro mais baixa, a seguir segue-se a imagem 4.1(b) e 4.1(a) com as suas taxas de erro muito perto uma da outra e por último a imagem 4.1(d). Para a imagem 4.1(d) este resultado justifica-se por esta ser uma imagem com muita homogeneidade dos seus objectos e consequentemente mais difícil de analisar e individualizar as várias regiões existentes sejam casas, ruas, estradas, carros ou árvores. Os resultados para imagem 4.1(c) justifica-se por duas razões, a primeira são as poucas regiões edificadas para identificar, pois é maioritariamente constituída por outras regiões e a

segunda razão é relativa ao contraste existente entre os edifícios e os restantes objectos da imagem. O resultado da imagem 4.1(b) que se destaca ligeiramente da imagem 4.1(a) mais uma vez por esta pertencer a uma zona rural onde as casas são mais facilmente identificadas, mas é de salientar que embora a imagem 4.1(a) pertença a uma zona urbana é constituída por algumas árvores o que justifica assim o resultado obtido que é bastante positivo tendo em conta as suas características.

Para a imagem 4.1(a), a selecção de 97% das características principais correspondeu a 4 características das 81 iniciais, por ser um valor reduzido resolveu-se seleccionar mais características e variar essa selecção entre 10, 20, 30, 40 e 50. Dos resultados obtidos verifica-se que a taxa de erro é menor para 40 características e como se pode ver pelo gráfico 4.2 para o valor de 50 e 30 características a diferença é bastante pequena em relação ao melhor resultado, mas para 97% que representam 4 características, 10 e 20 o resultado é bastante superior aos restantes o que prova que para a imagem 4.1(a) o melhor classificador necessita de entre 30 a 50 características para se obter uma taxa de erro mais baixa e conseqüente melhores resultados. A tabela 4.2 e o gráfico da imagem 4.3 mostram respectivamente a matriz confusão e a curva ROC para o resultado da imagem 4.1(a) com 40 características seleccionadas. Na matriz de confusão é possível verificar que obteve-se um melhor resultado para os verdadeiros negativos em relação aos verdadeiros positivos.

0	Positivo	Negativo
Positivo	85.809	10.657
Negativo	14.191	89.343

Tabela 4.2: *Matriz Confusão da imagem 4.1(a) obtida com extracção de 40 características*

Para a imagem 4.1(b) os resultados obtidos são semelhantes aos da imagem 4.1(a) com diferença para o melhor resultado, que desta vez revelou-se ser para 30 características, onde se obteve uma taxa de erro de 11.449. Na tabela 4.3 é possível ver a matriz confusão relativa ao teste feito com essas 30 características, onde a percentagem dos verdadeiros positivos é ligeiramente maior que a percentagem de verdadeiros negativos. Já na imagem

4.3 é possível ver a curva ROC para o melhor resultado da imagem 4.1(b) neste teste.

0	Positivo	Negativo
Positivo	89.225	11.604
Negativo	10.774	88.395

Tabela 4.3: *Matriz Confusão da imagem 4.1(b) obtida com extracção de 30 características*

Já para a imagem 4.1(c), como seria de esperar obteve-se os melhores resultados, pois esta como já foi dito apresenta contrastes acentuados entre as poucas regiões existentes o que facilita o desempenho do classificador. Em termos de características o melhor resultado vai para as 40 características embora para as outras características os resultados tenham sido bastantes semelhantes como se pode verificar no gráfico 4.2. Na tabela 4.4 e na figura 4.3 encontra-se respectivamente a matriz de confusão e o gráfico da curva ROC para o melhor resultado da imagem 4.1(c) quando variado o número de características.

0	Positivo	Negativo
Positivo	96.099	8.647
Negativo	3.900	91.352

Tabela 4.4: *Matriz Confusão da imagem 4.1(c) obtida com extracção de 40 características*

Por último temos a imagem 4.1(d) com os piores resultados, sendo que para esta, mais uma vez o resultado com 40 características seleccionadas foi o que obteve a taxa de erro mais baixa com 15.062. De salientar a diferença entre este resultado com o pior resultado de 30.084 que foi obtido com 97%, isto é 3 características seleccionadas, o que revela que para esta imagem que como já foi dito apresenta uma homogeneidade elevada entre os vários objectos presentes nesta, são necessárias 40 características para se obter os melhores resultados. Na tabela 4.5 e na figura 4.3 demonstra-se a respectiva matriz de confusão e gráfico da curva ROC para o melhor

resultado da imagem 4 quando se varia o número de características.

0	Positivo	Negativo
Positivo	76.45	12.755
Negativo	23.549	87.244

Tabela 4.5: Matriz Confusão da imagem 4.1(d) obtida com extracção de 40 características

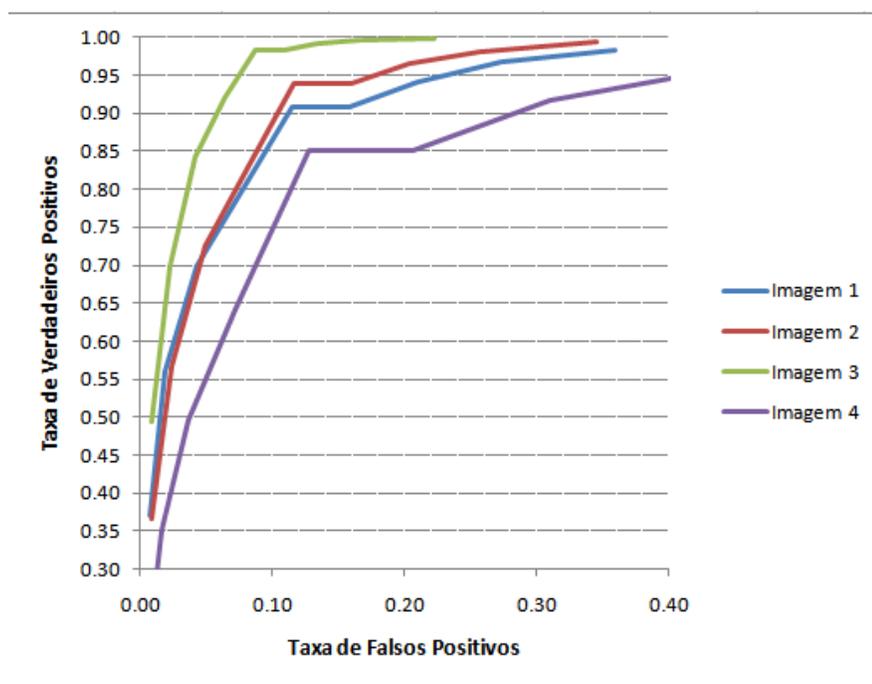


Figura 4.3: Curvas ROC para os melhores resultados

Para finalizar, conclui-se que para a variação do número de características o melhor resultado é maioritariamente para 40 características, sendo que este obteve os melhores resultados em três de quatro imagens aqui discutidas, o que se conclui que para se obter os melhores resultados é necessário escolher pelo menos 30 a 40 características principais.

De seguida discute-se os resultados para a variação do número de neurónios, sendo que os testes foram feitos para as quatro imagens da figura 4.1 e todos testes foram feitos tendo em conta os melhores resultados obtidos no parâmetro anterior, ou seja para a imagem 4.1(a), 4.1(c) e 4.1(d) testou-se com 40 características e para a imagem 4.1(b) testou-se 30 características.

Na tabela 4.6 apresenta-se os resultados relativos aos testes efectuados, bem como no gráfico da figura 4.4 mostra-se a distribuição desses mesmos resultados. Como seria de esperar os resultados entre as várias imagens mantêm-se iguais aos obtidos nos testes explicados anteriormente, ou seja, a imagem 4.1(c) continua com o melhor resultado e a imagem 4.1(d) com pior. Com estes testes pretendeu-se investigar qual o número de neurónios na camada intermédia que obtêm melhores resultados. Para isso fez-se variar esse número entre $k/2$, $k-10$, k , $k+10$ e $k*2$, tendo k valor igual ao número dos neurónios da camada de entrada que correspondem ao número de características seleccionadas para fazer o treino da rede neuronal, ou seja, temos $k=40$ para as imagens 4.1(a), 4.1(c) e 4.1(d) e $k=30$ para a imagem 4.1(b).

Imagens	Número de neurónios da camada intermédia				
	$k/2$	$k-10$	k	$k+10$	$k*2$
1	12	12.245	11.618	11.992	12.267
2	11.697	11.365	11.416	11.684	11.446
3	8.57	8.266	8.185	8.11	8.897
4	14.739	14.572	15.062	15.45	15.331

Tabela 4.6: Resultados das taxas de erro consoante a variação do número de neurónios da camada intermédia da rede neuronal

Em termos de comparação de resultados entre as imagens não existe um número de neurónios que se destaque mas sim o intervalo de $k-10$ até $k+10$, quer isto dizer que os melhores resultados obtêm-se para um número de neurónios da camada intermédia da rede neuronal em volta de k , pois à medida que se afasta este número de k a tendência é piorar os resultados.

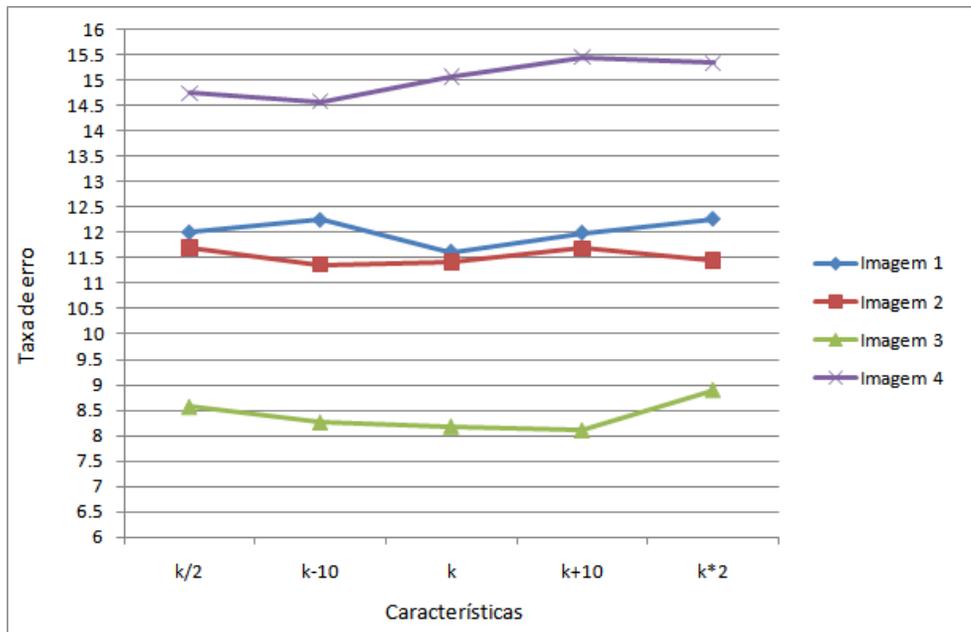


Figura 4.4: Gráfico das taxas de erro para a variação do número de neurónios na camada intermédia

Para a imagem 4.1(a) obteve-se o melhor resultado para o número de neurónios igual a k e o pior para k^*2 , o que prova precisamente a dedução anterior. A tabela 4.7 mostra a matriz de confusão para o seu melhor resultado onde se verifica um alto valor de verdadeiros negativos detectados. Já a curva ROC que expressa a variação da taxa de falsos positivos com a taxa de verdadeiros positivos encontra-se no gráfico da figura 4.5, sendo que a curva respeitante a esta imagem é a assinalada na legenda por imagem1.

0	Positivo	Negativo
Positivo	85.809	10.657
Negativo	14.191	89.343

Tabela 4.7: Matriz Confusão do melhor resultado da imagem 4.1(a) tendo em conta o segundo parâmetro

Para a imagem 4.1(b) e 4.1(d) o melhor resultado aconteceu para $k-10$ e para a imagem 4.1(c) o melhor resultado aconteceu para $k+10$. Quanto aos piores resultados, para comprovar a tendência de quanto mais afastado de k maior é a taxa de erro, os piores resultados das imagens 4.1(b) e 4.1(c) são $k/2$ e $k*2$ respectivamente. Por último o pior resultado para a imagem 4.1(d) foge a esta tendência pois acontece para $k+10$.

No gráfico da figura 4.5, e possível encontrar as curvas ROC para os melhores resultados expressos no parágrafo anterior e nas tabelas 4.8, 4.9 e 4.10 encontram-se as matrizes confusão, das imagens 4.1(b) e 4.1(c) e 4.1(d) dos melhores resultados obtidos tendo em conta este segundo parâmetro.

0	Positivo	Negativo
Positivo	88.791	11.208
Negativo	11.401	88.598

Tabela 4.8: Matriz Confusão do melhor resultado da imagem 4.1(b) tendo em conta o segundo parâmetro

0	Positivo	Negativo
Positivo	96.065	8.621
Negativo	3.9349	91.378

Tabela 4.9: Matriz Confusão do melhor resultado da imagem 4.1(c) tendo em conta o segundo parâmetro

0	Positivo	Negativo
Positivo	76.991	12.780
Negativo	23.008	87.219

Tabela 4.10: Matriz Confusão do melhor resultado da imagem 4.1(d) tendo em conta o segundo parâmetro

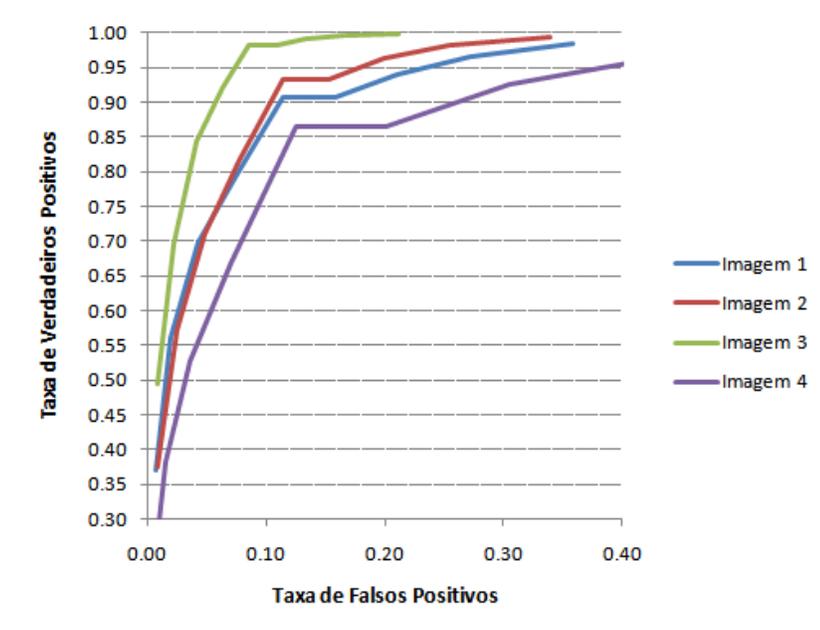


Figura 4.5: Curvas ROC dos melhores resultados da Tabela 4.6

4.2 Resultados para o conjunto de teste

Nesta secção apresentam-se alguns resultados obtidos de testes feitos às redes neuronais treinadas na fase anterior, a partir de imagens do conjunto de teste. Foram seleccionadas quatro imagens para realizar as simulações que a seguir serão apresentados, sendo estas as imagens da figura 4.6.

Os resultados obtidos são imagens com o mesmo tamanho que a imagem testada, mas somente constituídas por duas cores, preto e branco, onde a branco encontra-se as regiões identificadas pelo classificador como sendo regiões edificadas e a preto todas as regiões que não interessam.

As redes neuronais utilizadas para simular os resultados foram aquelas que logicamente obtiveram melhores resultados durante a fase de treino, sendo que esses melhores resultados encontram-se expressos ao longo da secção anterior e as suas curvas ROC estão representadas no gráfico da figura 4.5.



(a) Imagem 1



(b) Imagem 2



(c) Imagem 3



(d) Imagem 3

Figura 4.6: 3 imagens do conjunto de teste

Para cada imagem utilizou-se a rede que mais se apropriava a esta, tendo em conta as características de textura, da zona a que estas pertencem bem como dos objectos que estas contêm, ou seja, para testar uma imagem relativa a uma zona mais rural onde existe mais zonas verdes envolventes às casas, utilizou-se redes treinadas a partir de imagens com estas mesmas características, para assim obter um resultado mais satisfatório.

Em termos práticos, para as imagens 4.6(a) e 4.6(c) utilizou-se a rede

neuronal que obteve melhor resultado durante a fase de treino para a imagem 4.1(c) do conjunto de treino, pois estas são típicas imagens de zonas rurais e muito semelhantes entre si. Os resultado das taxas de erro encontram-se expressos na tabela 4.6, bem como no gráfico da figura 4.5, onde se pode verificar que a taxa de erro utilizada foi de 8.11.

Já para as imagens 4.6(b) e 4.6(d) utilizou-se a melhor rede neuronal obtida a partir do treino feito com a imagem 4.1(a), sendo que aqui optou-se mais uma vez por utilizar a rede neuronal obtida no treino de uma imagem com características semelhantes ao nível da zona representadas por estas que neste caso refere-se a uma zona mais urbana e consequente mais difícil de classificar como já foi referido anteriormente.

Para a imagem 4.6(a), a imagem obtida como resultado encontra-se na figura 4.7(b). Observando este resultado conclui-se que este é bastante satisfatório, sendo que as regiões da imagem que contêm casas, foram maioritariamente assinaladas e por outro lado são também poucos os casos em que regiões que não são compostas por edifícios foram assinaladas como tal. Ainda para este resultado pode-se afirmar como menos positivo o facto de as casas não serem identificadas individualmente umas das outras, mas sim as regiões por estes compostas, isto é, o resultado mostra-se positivo se tivermos em consideração a identificação de um aglomerado de casas e não individualmente. Este resultado foi dos melhor obtidos dos quatro aqui expostos, o que se conclui mais uma vez que estas imagens pelo seu tipo de composição, podem ser consideradas mais fáceis de classificar, pelas diversas razões já enunciadas ao longo deste documento.

A segunda imagem testada foi a imagem 4.6(b), e o resultado obtido encontra-se na figura 4.8(b). Como se pode ver este resultado é bastante positivo pois o seguimento das casas foi assinalado com sucesso e por outro lado a área florestal presente na imagem foi preenchida a preto, o que representa que o classificador consegue identificar bem o contraste entre estas duas regiões da imagem. Outro aspecto positivo deste resultado é o facto de as regiões entre os edifícios, quando são ruas ou estradas, não terem sido assinalados como sendo casas, pois este é um dos aspectos que torna difícil a classificação de imagens de zonas urbanas, portanto para esta imagem conclui-se que a rede neuronal utilizada para a classificar superou as expectativas tendo em conta a dificuldade da imagem por pertencer a

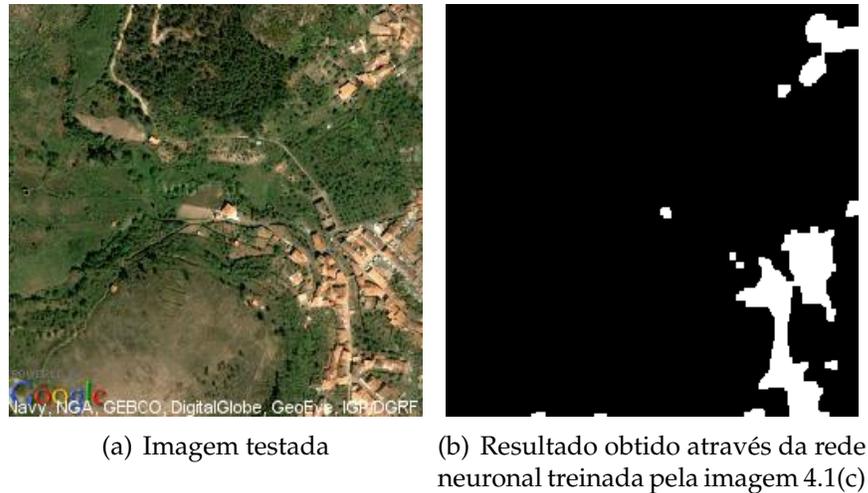


Figura 4.7: Resultados obtidos para a imagem 4.6(a)

uma zona urbana e consequentemente apresentar texturas mais idênticas e susceptíveis de enganar o classificador na tomada de decisão.

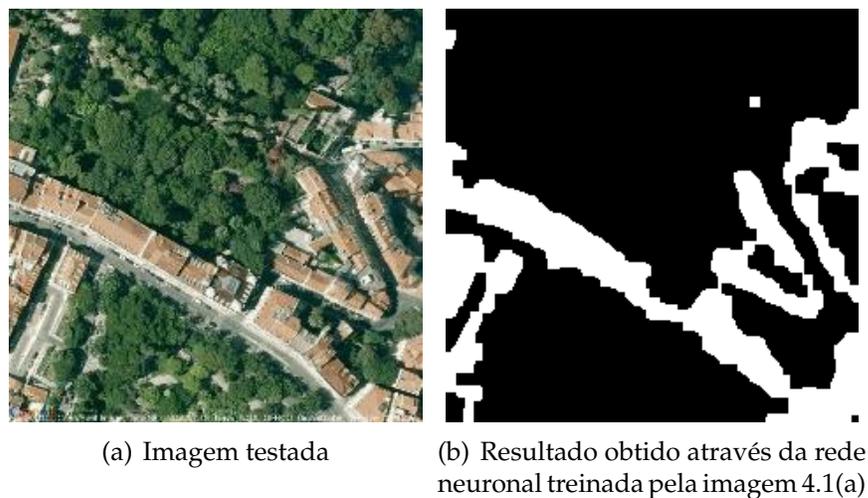


Figura 4.8: Resultados obtidos para a imagem 4.6(a)

No terceiro exemplo temos o resultado obtido representado na figura 4.9. Este resultado é bastante satisfatório pois se apreciarmos o tipo de ima-

gem que esta representa conclui-se que este resultado surpreendeu pela positiva ao conseguir identificar as muitas regiões edificadas existentes na imagem e também delinear bem os contornos destas, em situações de dificuldade por esta imagem apresentar algumas propriedades já referidas como típicas de imagens de zonas mais urbanas e conseqüentemente mais difíceis.



(a) Imagem testada

(b) Resultado obtido através da rede neuronal treinada pela imagem 4.1(d)

Figura 4.9: Resultados obtidos para a imagem 4.6(a)

Estas três primeiras imagens foram as que apresentaram melhores resultados nos testes efectuados, de seguida apresenta-se um resultado não tão positivo por a imagem testada apresentar algumas características próprias que as torna mais difícil de classificar.

Para a imagem 4.6(c), constata-se que o resultado apresenta algumas falhas com se pode ver através da figura 4.10(b), onde estas são visíveis, não tanto ao nível de não identificar correctamente as regiões onde existem casas, mas por outro lado ao identificar zonas onde estas não existem. Este resultado justifica-se pois a imagem testada, apesar de ser de um meio rural, contém na zona envolvente às casas texturas e cores que induzem o classificador em erro.



(a) Imagem testada



(b) Resultado obtido através da rede neuronal treinada pela imagem 4.1(c)

Figura 4.10: Resultados obtidos para a imagem 4.6(c)

Capítulo 5

Conclusão e trabalho futuro

5.1 Conclusão

Os objectivos propostos deste projecto foram atingidos com sucesso, pois através da extracção de características de textura em uma imagem foi possível identificar zonas edificadas.

Os resultados obtidos neste projecto foram bastante satisfatórios, pois a análise de texturas revelou-se eficaz para identificar regiões construídas pelo ser humano. Os melhores resultados foram obtidos do conjunto de dados referente às imagens com regiões menos homogéneas, por exemplo em zonas rurais onde o contraste das casas com o resto é mais acentuado, porque neste meio normalmente à volta das casas existe alguma vegetação, florestas ou áreas agrícolas, o que vai facilitar na tomada de decisão ao detector. Já em áreas urbanas os resultados, embora positivos, revelam-se menos precisos pois as imagens típicas destas zonas contêm por exemplo estradas, ruas e carros o que torna bastante difícil a tomada de decisão para o detector.

Alguns dos resultados demonstrados serviram para concluir que quanto maior for a distância para a superfície terrestre, pior será para os classificadores identificarem correctamente as regiões pretendidas. Por outro lado esta altitude também não pode ser muito reduzida pois a área testada seria muito menor o que levaria a um custo computacional elevado, caso se pretenda analisar uma área muito grande, pois levaria a ter de analisar

muitas imagens. Esta conclusão encontra-se bem patente no resultado da imagem presente na figura 4.10 onde os resultados não foram os melhores precisamente por esse factor, já por outro lado o resultado das imagens 4.7 e 4.8 permitem concluir que a uma altitude certa consoante o tipo de imagem os resultados obtidos são bastante positivos, pois para a primeira imagem que se encontra a uma altitude considerável obteve-se uma boa classificação, pois a imagem apresenta um contraste entre casas e áreas agrícolas ou florestais, o que facilita no processo de detecção. Por outro lado para a segunda imagem a altitude já se encontra mais reduzida e tendo em conta o resultado esta redução revela-se importante.

5.2 Trabalho Futuro

Numa óptica de trabalho futuro pode-se adicionar alguns aspectos que tornem o sistema automático de identificação de regiões edificadas mais poderoso e capaz de obter bons resultados em imagens com um grau de homogeneidade muito elevado.

Um dos aspectos é acrescentar um conjunto de características mais focalizadas em identificar áreas urbanas, pois como já foi referido, o método implementado revelou-se mais eficaz em imagens de zonas mais rurais onde o contraste entre as zonas edificadas e o resto é mais acentuado, por isso a aplicação de novas características que sejam capazes de identificar individualmente as casas e edifícios de por exemplo estradas, ruas, carros, com mais eficiência para poder então obter melhores resultados em imagens de zonas urbanas onde estas propriedades específicas se encontram mais visíveis. A implementação de novas características poderá também permitir a identificação de casas e edifícios isoladamente, e assim ser possível contar o número de casas numa certa região.

Outro aspecto de possível implementação no futuro, é um método que permita testar as imagens a um nível de altitude diferente e com isso conseguir obter para cada imagem o nível de altitude que melhor se adequa a esta, isto é, dependendo do tipo de imagem e do tipo de objectos que estas contêm conseguir definir uma altitude onde se obtêm os melhores resultados.

Bibliografia

- [1] Neuro AI, *Intelligent systems and neural networks*, 2007, <http://www.learnartificialneuralnetworks.com/>.
- [2] Google Code, <http://code.google.com/intl/pt/apis/maps/documentation/staticmaps/>.
- [3] C.M. BROWN Dana H. Ballard, *C.m. computer vision*, Nova Jersey: Prentice Hall, 1982.
- [4] J. G. Daugman, *Uncertainty relation for resolution in space, spatial frequency and orientation optimized by two-dimensional visual cortical filters*, *Journal of Optical Society of America* **2** (1985), no. 7, 1160–1169.
- [5] D. Gabor, *Theory of communication*, *Journal of the Institute of Electrical Engineers* **93** (1945), 429–457.
- [6] K. Shanmugan Haralick, R.M. and I. Dinstein, *Textural fetures for image classification*, *IEEE Transactions on Systems, Man and Cybernetics* **3** (1973), no. 6, 610–621.
- [7] Ming-Kuey. Hu, *Visual pattern recognition by moment inoariants*, *IRE Trans. on Information Theory* (1962), 179–187.
- [8] Duin R. P. W. e Mao Jain, A. K., *J. statistical pattern recognition*, *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2000), 4–37.
- [9] Kendall, *J. statistical pattern recognition*, 1980.
- [10] Richard E. Woods Rafael C. Gonzalez and Steven L. Eddins, *Digital image processing using matlab*, Pearson Education, 2004.

- [11] M. A. Shaban and Dikshit, *Textural classification of high resolution digital satellite imagery*, IEEE Transactions on Computers (1998), 2590–2592.
- [12] Prof S K Shah and V Gandhi, *Image classification based on textural features using artificial neural network (ann)*, IE (I) Journal-ET **84** (2004), 72–74.
- [13] Jamie Shutler, *Statistical moments*, 2002, http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/SHUTLER3/node1.html.
- [14] Smith, L. *i. a tutorial on principal component analysis*, 2002.
- [15] Wiki, *Rede neural*, Abril 2010, http://pt.wikipedia.org/wiki/Rede_neural.
- [16] Wikipédia, *Hsl and hsv*, 2010, http://en.wikipedia.org/wiki/HSL_and_HSV.