Reconhecimento de Imagens Aéreas

Hugo Drumond
hugo.drumond@fe.up.pt
Inês Caldas
up200904082@fe.up.pt
Pedro Moura
up201306843@fe.up.pt

Faculdade de Engenharia da Universidade do Porto
s/n, R. Dr. Roberto Frias, 4200-465
Porto, Portugal

Abstract

A classificação de imagens áreas atraiu muita atenção em inúmeras aplicações de . Uma classificação de imagens com sucesso depende de vários fatores, desde a disponibilidade e complexidade dos dados à existência de algoritmos de classificação adequados, entre muitos outros. Devido à grande diversidade de cenas aéreas, com grandes variações entre imagens da mesmas classe, o processo de classificação torna-se bastante desafiador. Não existe um único método de classificação que seja o melhor e neste artigo implementamos três diferentes métodos de classificação e discutimos as vantagens e limitações das várias implementações. TODO

1 Introdução

A análise de imagens de satélite sempre recebeu um grande interesse tanto pela parte dos académicos como das indústrias, devido às suas inúmeras aplicações, por exemplo, na comunidade de *remote sensing*. No entanto, a classificação e compreensão de cenas aéreas acarretam muitos desafios técnicos, tais como a grande diversidade de classes e detalhes obscuros nas imagens. Nos últimos anos, os dados de imagens de alta resolução, fornecidos por novos e avançados sensores espaciais, abriram novas oportunidades para caracterizar as cenas aéreas com bases nos padrões espaciais e estruturais codificados nas imagens. Impulsionado pelo drástico crescimento de imagens aéreas de alta resolução, da publicação de vários *datasets* e de novos avanços computacionais, o problema de classificação de imagens tem vindo a recuperar atenção e tem se visto a publicação de inúmeros algoritmos para a classificação de imagens aéreas, que vão desde *machine learning* a abordagens orientadas por dados.

A classificação robusta das imagens de satélite, que visa rotular automaticamente uma imagem aérea com uma categoria semântica, é um passo fundamental para a compreensão destas imagens. Neste artigo, vamos estudar três métodos de classificação de imagens, comparando os resultados obtidos e discutindo as limitações e vantagens de cada um. Os três métodos escolhidos foram o histograma de cor; *bag of words* usando *sift* e uma CNN. Para treinar e classificar cada um dos modelos, recorreu-se ao dataset AID Xia et al. [4], que será descrito na secção 4.1.

2 Classificação de Imagens Aéreas

A classificação de imagens é um processo complexo e requer que vários fatores sejam tomados em consideração. Os principais passos na classificação de imagens podem incluir a determinação do sistema de classificação apropriado, a seleção do conjunto de treino, pré-processamento das imagens, método de extração das características das imagens, processamento pós-classificação e escolha das medidas de avaliação de precisão e robustez.
Segundo Xia et al. [4], os métodos de classificação de imagens podem ser divididos em três categorias principais: métodos que usam características visuais de baixo nível, métodos que dependem de representações visuais de médio nível e métodos baseados em informação visual de alto nível.

Os métodos de baixo nível tentam classificar as imagens com base em características visuais de baixo nível, como por exemplo, espectro de cor, textura, estrutura, etc. Logo, a imagem é descrita por um vetor definido pelas atributos visuais extraídos quer a nível local, como global.

Contrariamente aos métodos anteriores, os algoritmos de médio nível tentam desenvolver uma representação holística sobre a imagem, através da procura de padrões estatísticos sobre as características locais extraídas das imagens.

Atualmente, os métodos de *deep learning* têm alcançado resultados impressionantes em muitos domínios de visão por computador, incluindo na classificação de imagens. Quando comparado com os métodos de baixo e médio nível, as estratégias de *deep learning* são capazes de aprender semânticas mais abstratas e discriminativas sobre as características extraídas da imagens e alcançar muito melhores resultados. Para uma análise mais completa os vários métodos de classificação e uma análise mais completa sobre estes métodos referimos Xia et al. [4].

3 Sistema Desenvolvido

3.1 Histograma de Cores

Dado um espaço de cor discreto, definido por um conjunto de eixos de cor (por exemplo, vermelho, verde e azul), o histograma de cor é obtido a partir da discretização das cores da imagem e da contagem do número de vezes que cada cor "discreta" surge no array da imagem. Os histogramas de cor são invariantes à translação e rotação dos eixos de visão, mas são sensíveis a variações da mudança do ângulo de visão, variações em escala e na presença de oclusões. Para ser possível identificar uma imagem com base no seu histograma de cor, é necessário avaliar o grau de semelhança do seu histograma com os histogramas de cor na base de dados. Para tal, usou-se a distância de *Hellinger*. Enquanto que a distância euclidiana foca-se na diferença absoluta, a distância de *Hellinger* é sensível a diferenças relativas e, embora simples, é eficaz o suficiente para diferenciar os contrastes nas estruturas locais significativas de uma imagem. A distância de *Hellinger*, para duas distribuições discretas de probabilidades $P(p_1,...,p_k)$ e $Q(q_1,...,q_k)$ é definida como:

$$H(P,Q) = \frac{1}{\sqrt{2}}\sqrt{\sum_{i=1}^{k}\left(\sqrt{p_i}-\sqrt{q_i}\right)^2}, \quad (1)$$

3.2 *Bag of Words*

O modelo BOV (*Bag of Words*), proposto inicialmente por Csurka et al. [1] e Sivic and Zisserman [3], é essencialmente um modelo que permite criar um vocabulário que melhor descreve uma imagem em termos das características extrapoladas. A sua implementação pode ser descrita em quatro passos:

- Determinar as características de uma imagem de uma determinada categoria
- Construir um vocabulário visual por *clustering*, seguido por uma análise de frequência
- Classificação das imagens com base no vocabulário gerado
- Obtenção da classe "óptima" para a imagem de teste

Bag of Visual Words é um modelo de aprendizagem supervisionado que necessita de um conjunto de imagens para treinar o modelo e outro conjunto para testar a implementação. Para extrair as características das imagens, pode-se usar algoritmos como o SURF ou o SIFT. Define-se um *patch* em torno de cada um dos pontos retirados e calculou-se o *Histogram of Oriented Gradients*(HoG) para cada *patch*. Recolhida esta informação é necessário agrupar características similares em grupos e definir as "palavras" do nosso vocabulário. O método usado para agrupar as características, foi o algoritmo *k-means clustering*. Vamos supor

que existem X objetos que podem ser divididos em K *clusters* e que o input pode ser definido por um conjunto de características $X = x_1, x_2, ...x_n$, o objetivo deste algoritmo é minimizar a distância entre cada ponto e os centroides definidos:

$$\arg\min_{S} \sum_{i=1}^{k} \sum_{x \in S_i} \|\mathbf{x} - \mu_i\|^2 \quad (2)$$

onde μ é a média de pontos em cada *cluster* S_i e S representa o conjunto de pontos divididos pelos *clusters* $S_1, S_2, ...S_i$ Os centroides resultantes do *clustering* são tratados como as palavras visuais do dicionário. Assim, cada característica de uma imagem é mapeado para uma determinada palavra e imagem pode ser representada pelo histograma das palavras que constituem o vocabulário. Obtidos os histogramas, é necessário treinar uma SVM (*Support Vector Machine*) sobre estes e comparar com o conjunto de imagens de teste. Casos os resultados não se provem satisfatórios, é possível ajustar o conjunto de imagens de treino e afinar os parâmetros da SVM.

3.3 *Convolution Neural Network*

Os métodos que têm garantido resultados mais robustos na classificação de imagens, são métodos de *deep learning*. Para ser possível comparar os resultados obtidos com métodos mais tradicionais, implementou-se a framework RetinaNet com o model ResNet50, descrita em Lin et al. [2], usando a API *Keras*. A arquitetura ResNet é uma arquitetura extremamente profunda mas com uma complexidade inferior às redes que a precederam. Com a Resnet introduziu-se uma nova framework, à base de *residual learning*, que facilita o treino destas redes tão profundas, que passaram a ser conhecidas por *residual networds*. O R*esidual learning* envolve *residual functions*. Admitamos que um pequeno conjunto de *layers* pode aproximar uma função complexa $F(x)$ onde x é o input da primeira *layer*, podemos dizer também que pode aproximar a rede residual $F(x) - x$. Logo, esse conjunto de *layers* aproxima a função residual $G(x) = F(x) - x$ e a função original torna-se $G(x) + x$. Embora ambas as funções sejam capazes de aproximar a função desejada, a facilidade de treinar a função residual é maior. Estas funções residuais são encaminhadas ao longo das layers da rede usando *identity mapping shortcut connections*. A ResNet50 possui 50 layers e uma típica aquitetura ResNet consiste em filtros $3x3$ e $1x1$, *pooling layers* e *residual conections* e com uma única *softmax layer* no final

4 Experiências

Foram conduzidas várias experiências para testar o treino e a precisão dos vários métodos implementados. As experiências foram conduzidas com recurso ao *dataset* que se encontra descrito na secção seguinte. Para realizar as experiências foi necessário dividir as imagens em dois conjuntos, um de treino e outro de teste. A divisão foi feita segundo o seguinte critério: 200 imagens de cada classe pertencem ao conjunto de treino e as restantes imagens por classe pertencem ao conjunto de teste. Desta forma, as percentagens de imagens usadas como treino, encontram-se descritas na tabela 1.

Table 1: Percentagem de imagens usadas no conjunto de treino por classe.

Class	%	Class	%	Class	%
airport	56	farmland	54	port	53
bare land	65	forest	80	railway station	77
baseball field	91	industrial	51	resort	69
beach	50	meadow	71	river	47
bridge	56	medium res.	69	school	67
center	77	mountain	59	sparse res.	67
church	83	park	57	square	61
commercial	57	parking	51	stadium	69
dense res.	49	playground	54	storage tanks	56
desert	67	pond	48	viaduct	48

4.1 Dataset

Como referido anteriormente, para testar a implementação dos métodos desenvolvidos recorreu-se ao *dataset* AID. Este *dataset* é composto por imagens em grande-escala recolhidas no *Google Earth* que se encontram divididas em 30 classes diferentes: *airport, bare land, baseball field, beach, bridge, center, church, commercial, dense residential, desert, farmland, forest, industrial, meadow, medium residential, mountain, park, parking, playground, pond, port, railway station, resort, river, school, sparse residential, square, stadium, storage tanks, viaduct*. O *dataset* é composto por um total de 10.000 imagens, divididas heterogeneamente pelas 30 classes. Todas as imagens têm um tamanho fixo de 600x600 pixeis e portanto cobrem áreas com diferentes resoluções [1]

4.2 Resultados

Para medir a qualidade dos métodos, foi calculada a exatidão dos resultados obtidos. A exatidão define-se como:

$$ACC = \frac{TP + TN}{P + N}, \quad (3)$$

onde a soma de TP com TN é o número de imagens classificadas corretamente e a soma de P com N o número total de imagens testadas. Com o método do histograma de cor, obteu-se uma exatidão de 60.15%. Quando comparado com a exatidão obtida no artigo Xia et al. [4], que foi de 37.28, para um conjunto de treino de 50%, pode-se afirmar que os resultados obtidos foram bastante satisfatórios.

Para o método BoVW, optou-se por usar o algoritmo *SURF* em vez de *SIFT*, dado que este último é computacionalmente mais exigente e para a carga dos dados usados levou a problemas de memória, embora este produza pontos característicos mais robustos. A exatidão obtida foi de 55.04. Embora para os conjuntos de treino e teste na proporção de 60% e 40% a exatidão obtida sobe para 64.29. Comparando estes resultados novamente com os obtidos no artigo Xia et al. [4] que obtiveram uma exatidão de 68.37 os resultados não são tão satisfatórios. Contudo, isto pode explicar-se pelo fato de terem usado o método *SIFT* para extrair os pontos característicos em vez do método *SURF*, que é bastante mais robusto. Pela análise das *confusion matrix* verifica-se que o número de previsões corretas por classe se encontra relacionado com a percentagem do números de imagens usadas para treino, e que quanto maior for este número, melhor será a exatidão na previsão para essa classe. Algumas discrepâncias que possam surgir para o método de BoVW devem-se a problemas na deteção de pontos característicos.

Implementou-se o sistema de treino e classificação para o ResNet50[2]. Contudo, devido a restrições computacionais não foi possível treinar a rede para além de duas *epoch*. Para este treino, os melhores *scores* obtidos rondaram valores entre os 0.2. Para estes valores, a rede não é capaz de produzir resultados fiáveis, só a partir de um *score* de 0.5, o que na realidade se confirmou, pois para o melhor valor obtido para cada imagem de teste, a previsão de classificação encontrava-se errada. Para se obterem resultados fiáveis e robustos, o aconselhável seria treinar a rede pelo menos 50 *epochs*, mas de preferência para valores acima dos 100.

5 Conclusões

Através da implementação dos três diferentes métodos, podemos verificar que a classificação de imagens não é uma tarefa trivial. A grande variedade nas características das imagens pertencentes à mesma classe torna difícil uma classificação robusta. Além disso, o grande volume de dados necessários de processar e analisar, exige um grande poder computacional que restringiu o trabalho desenvolvido. Mesmo contra estas restrições, foi possível obter resultados satisfatórios para os métodos do histograma de cores e BoVW. Com tempo suficiente, seria interessante concluir o treino da rede ResNet50 e verificar a exatidão obtida, sendo de esperar que esta fosse bastante elevada. Além disso, esta rede é capaz de segmentar uma imagem e classificar diferentes segmentos de acordo com a classe a que pertencem.

[1] Para uma descrição mais detalhada do *dataset* consultar Xia et al. [4]

[2] baseado no trabalho que se encontra em https://github.com/fizyr/keras-retinanet

References

[1] Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *In Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004.

[2] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007, Oct 2017. doi: 10.1109/ICCV.2017.324.

[3] J. Sivic and A. Zisserman. Video google: a text retrieval approach to object matching in videos. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 1470–1477 vol.2, Oct 2003. doi: 10.1109/ICCV.2003.1238663.

[4] G. S. Xia, J. Hu, F. Hu, B. Shi, X. Bai, Y. Zhong, L. Zhang, and X. Lu. Aid: A benchmark data set for performance evaluation of aerial scene classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(7):3965–3981, July 2017. ISSN 0196-2892. doi: 10.1109/TGRS.2017.2685945.

CH

Actual \ Predicted	storage tanks	commercial	pond	square	bareland	resort	sparse res	mountain	farmland	bridge	meadow	baseball field	desert	center	port	beach	dense res	school	playground	stadium	parking	forest	river	medium res.	park	industrial	viaduct	airport	railway station	church
storagetanks	113	5	0	3	2	1	0	0	0	1	0	1	0	0	0	1	6	0	0	1	6	0	0	1	1	5	3	4	4	2
commercial	2	114	0	1	1	1	0	0	0	0	0	0	0	0	0	0	7	4	0	0	1	0	0	0	4	4	2	1	6	2
pond	6	1	73	5	3	3	24	11	5	13	12	5	0	1	7	2	3	2	2	0	0	2	11	8	9	5	5	0	2	0
square	1	3	0	62	1	4	2	0	1	1	0	2	0	6	0	0	10	3	0	2	1	0	0	5	5	6	5	3	4	3
bareland	2	0	0	0	93	0	0	3	1	0	0	0	5	0	0	0	0	0	0	0	4	0	0	0	0	0	0	1	1	0
resort	0	1	0	0	1	71	0	0	0	1	0	0	0	0	0	0	2	1	0	0	0	1	0	1	5	0	0	0	5	1
sparse res.	0	0	0	0	0	0	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	5	0	0	0	0	0	0
mountain	0	1	1	0	0	0	2	108	4	0	4	1	1	1	0	0	2	0	0	0	1	4	2	1	0	1	6	0	0	0
farmland	2	1	0	1	4	2	1	7	100	0	12	2	1	0	0	1	2	1	1	0	0	3	11	4	5	3	3	0	2	1
bridge	2	10	5	2	0	7	0	3	6	49	5	1	2	1	20	0	3	2	8	0	1	2	1	0	10	2	13	0	4	1
meadow	0	0	0	0	0	0	2	2	1	0	68	1	0	0	0	0	0	0	4	0	0	1	1	0	0	0	0	0	0	0
baseball fiel	0	0	0	0	0	0	0	0	0	0	1	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
desert	0	0	0	0	13	0	0	0	0	0	0	0	86	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
center	3	5	0	2	0	1	0	0	0	0	0	0	0	18	0	0	2	8	0	0	1	0	0	0	1	7	3	0	2	7
port	0	4	4	1	0	9	0	1	0	14	0	0	0	1	110	8	0	1	4	0	1	1	0	0	4	1	6	1	8	1
beach	3	1	1	3	4	6	0	3	3	1	3	0	0	1	17	122	1	1	2	2	2	0	3	1	4	6	1	4	5	0
dense res.	1	4	0	0	0	2	0	0	1	1	0	0	0	0	0	0	163	7	0	0	0	0	0	0	5	2	2	2	9	11
school	3	3	0	2	0	3	0	1	0	0	0	0	0	0	0	0	6	67	0	0	0	0	0	1	7	0	3	0	2	2
playground	0	1	0	5	0	8	2	2	4	8	5	7	0	2	3	2	11	8	56	2	2	1	3	9	17	4	3	1	3	1
stadium	3	6	0	5	0	1	0	0	0	0	0	0	6	0	0	0	9	11	1	29	1	0	0	2	2	2	4	1	6	1
parking	4	1	0	0	0	1	0	0	1	1	0	0	1	2	0	0	8	0	0	1	149	0	0	1	1	7	3	2	2	5
forest	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	46	0	0	1	0	0	0	0	0
river	1	0	2	1	1	1	7	17	16	2	14	1	0	0	1	0	4	2	0	0	1	12	110	1	5	3	4	2	1	1
medium res.	0	0	0	0	1	3	2	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	76	3	0	1	0	1	0
park	0	12	0	4	0	9	0	2	1	2	0	1	0	0	1	0	11	14	0	0	0	0	1	2	70	2	5	0	13	0
industrial	11	2	0	0	1	3	0	2	0	0	0	0	0	1	0	0	10	8	0	1	2	0	1	1	5	111	6	4	21	0
viaduct	13	7	1	5	1	4	2	2	1	1	0	0	0	2	0	0	18	7	1	0	1	0	1	8	16	16	97	6	7	3
airport	14	1	1	4	7	2	0	6	4	2	0	0	0	6	1	0	2	0	0	2	3	0	1	1	5	6	8	72	12	0
railway stati	1	9	1	2	0	5	1	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	3	4	2	2	25	0
church	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	2	0	0	0	0	35

BoVW

Actual \ Predicted	storage tanks	commercial	pond	square	bareland	resort	sparse res	mountain	farmland	bridge	meadow	baseball field	desert	center	port	beach	dense res.	school	playground	stadium	parking	forest	river	medium res.	park	industrial	viaduct	airport	railway station	church
storagetanks	126	2	0	2	0	1	1	0	0	0	2	2	0	2	2	1	1	2	1	0	1	0	1	0	1	2	2	5	0	3
commercial	0	110	1	2	1	0	0	0	0	0	0	0	0	0	9	0	2	6	0	0	0	0	2	0	1	3	1	0	6	6
pond	5	0	51	6	12	5	5	3	15	2	5	7	6	3	5	16	3	3	5	5	0	4	24	6	10	0	8	6	0	0
square	4	3	1	32	2	2	4	0	3	6	0	2	4	1	5	2	1	5	5	2	2	0	3	9	7	3	4	6	1	11
bareland	0	0	1	0	44	0	2	2	1	0	6	0	41	1	1	5	0	0	1	0	0	1	3	0	1	0	0	0	0	0
resort	2	5	0	5	3	24	1	0	1	2	0	2	0	1	3	2	1	5	1	5	0	0	4	4	7	2	3	2	1	4
sparse res.	0	0	3	0	0	1	81	0	0	0	2	1	0	0	0	0	0	0	3	0	0	2	0	7	0	0	0	0	0	0
mountain	1	0	0	0	2	0	0	131	2	1	0	0	2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
farmland	2	1	5	0	3	0	2	0	114	1	6	0	15	0	0	2	0	0	6	1	0	0	3	1	1	1	0	0	5	1
bridge	3	0	19	1	4	4	3	0	10	23	5	6	0	3	12	10	0	1	21	5	0	0	5	3	0	0	9	7	5	1
meadow	0	0	0	0	4	0	1	0	1	0	49	0	13	0	1	5	0	0	0	0	0	3	2	1	0	0	0	0	0	0
baseball fiel	2	0	2	1	0	0	0	0	0	1	0	11	0	0	0	0	0	1	0	0	0	0	0	2	0	0	0	0	0	0
desert	1	0	0	0	5	0	0	1	3	0	0	0	75	0	0	7	0	0	0	0	0	1	1	0	0	0	1	0	0	0
center	5	0	1	5	0	1	1	0	1	0	0	3	0	6	0	0	0	0	3	6	1	0	1	4	1	5	2	4	3	7
port	2	6	3	2	0	2	0	0	1	7	0	3	1	0	111	3	3	1	1	4	0	0	3	1	7	5	3	6	3	2
beach	1	0	3	3	14	1	0	5	2	0	23	3	19	0	0	110	0	0	2	1	2	2	4	2	0	0	0	1	1	1
dense res.	0	5	0	2	1	0	0	1	0	0	1	1	0	0	3	0	149	9	1	1	3	1	1	2	11	5	0	1	2	10
school	1	11	0	1	0	2	0	0	1	1	0	0	0	1	2	0	11	36	2	2	1	0	3	7	8	2	1	3	0	4
playground	1	0	3	7	9	3	3	0	15	12	0	13	2	5	5	7	0	3	56	6	0	0	1	4	0	0	3	10	0	2
stadium	7	0	0	1	0	1	0	0	2	1	0	3	0	0	2	1	1	2	6	38	0	0	0	0	0	9	1	11	2	2
parking	0	1	0	0	2	0	1	0	0	0	0	0	0	0	0	2	0	0	0	0	176	0	0	1	1	1	2	1	0	2
forest	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0	0
river	1	2	20	3	4	7	5	7	24	1	9	0	3	1	5	12	0	6	4	1	0	7	64	1	8	0	2	5	8	0
medium res.	0	0	1	1	2	1	5	0	0	1	1	4	4	0	1	2	0	4	0	0	1	1	0	53	0	0	0	0	1	7
park	9	8	1	3	0	5	0	1	0	0	0	4	0	0	7	0	3	3	0	0	0	0	6	0	89	1	4	0	6	0
industrial	3	4	1	5	1	5	1	0	1	0	0	2	1	3	3	1	9	6	1	13	2	0	2	2	3	89	3	3	24	2
viaduct	2	1	3	1	2	0	0	0	6	0	0	3	1	2	0	0	0	0	7	5	2	1	4	1	2	4	150	7	16	0
airport	5	3	3	2	3	3	0	0	6	4	0	0	1	2	2	0	0	0	1	13	2	0	3	1	1	10	3	86	4	2
railway stati	1	2	0	2	0	2	0	0	0	0	0	0	0	1	2	0	0	0	1	0	0	0	0	0	1	4	2	2	38	2
church	1	1	0	1	0	0	0	0	0	0	0	0	0	2	2	0	0	0	0	2	0	0	0	2	0	0	0	1	1	27