Detecção de carro em imagens

Carlos Frias
carlos.frias@fe.up.pt
Nuno Marques
nuno.marques@fe.up.pt

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

Abstract

Este relatório procura documentar a implementação de uma aplicação de visão por computador para a deteção de veículos automóveis em imagens e fornecer uma estimativa da sua localização na imagem. Na implementação da aplicação foi utilizada uma abordagem baseada em bag-of-words. A aplicação foi desenvolvida em C++ em ambiente Windows, utilizando a biblioteca OpenCV para visão de computador.

1 Introdução

Pretendeu-se implementar uma aplicação que analise um conjunto de imagens e que possa detetar a existência de carros nessas imagens. Para tal, o sistema foi baseado no treino e teste de classificadores para a distinção visual entre carros e outros conteúdos. A técnica utilizada foi baseada na abordagem de bag-of-words [2] [1]. Nesta abordagem a ideia básica é tratar as imagens como documentos procurando a ocorrência de determinadas palavras, isto pode ser conseguido seguindo três passos, a saber: A deteção dos pontos característicos, a descrição das características e geração das classes de objeto. Deste modo, a representação em bag-of-words quantifica o espaço de caraterísticas para criar um conjunto discreto de palavras visuais.

Os descritores são agrupados em clusters utilizando o método k-means que procura particionar n observações de entre k clusters onde cada observação pertence ao cluster mais próximo da média. Assim, são criados os dicionários para serem o vocabulário do extrator do bag-of-words.

Após a definição do vocabulário do bug-of-words são analisadas as imagens de treino para a construção de descritores para duas classes de objetos, carro e não carro. Para a divisão nessas duas classes são utilizadas máscaras contendo zonas a vermelho representando a parte visível do carro, verde para a parte do carro sujeita a oclusões e preto representado o ambiente de fundo da imagem. A figura 1 ilustra uma imagem e o template correspondente:

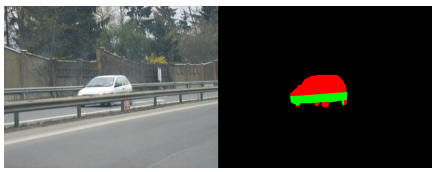

Figure 1: Imagem de treino e respetiva máscara.

Foram utilizados dois tipos de modelos de aprendizagens, o Normal Bayes Classifier e o Support Vector Machines Classifier. No teste de novas imagens, uma vez que a localização dos carros e o seu escalamento é indeterminado, foi necessário criar um método de varrimento das imagens com janelas de várias dimensões. Cada janela de varrimento é representada por um vetor bag-of-words testado com o modelo previamente treinado. O resultado final será a definição de áreas da imagem contendo um carro. A figura 2 ilustra o resuldado obtido para uma determinada imagem:

Figure 2: Resultado final na deteção de carros numa figura.

2 Implementação

Inicialmente é efetuada verificação da existência dos ficheiros das imagens de teste, treino e respetivas máscaras, caso não existam, o programa termina. Para carregar para um vetor os números identificadores das imagens de teste e de treino é efetuada a chamada a função readFileListFromFile criada para o efeito. É também, instanciado o detetor de pontos característicos, o descritor, o matcher, o treiner do Bag-of-words e o descritor do Bag-of-words.

Em seguida, passamos à geração do dicionário de palavras. Se este não existir em disco (se a flag estiver a false) então será criado o dicionário, percorrendo cada uma das imagens, detetando os pontos característicos e extraindo os descritores de cada imagem. Esses descritores são guardados num vetor/matriz.

Constrói-se o dicionário, efetuando-se um cluster k-means a esses descritores através da chamada à função cluster do treiner do Bag-of-words que é gravado em disco. Se o dicionário já existir (flag a true) então ele é carregado do disco. É, em seguida, atualizado o vocabulário do descritor do Bag-of-words com o dicionário criado ou carregado do disco, através da chamada à função setVocabulary.

Posto isto, segue-se o treino do classificador, para tal, é verificado se existe o ficheiro com a matriz de treino. Se não existir (flag a false), então será criado esse ficheiro. Deste modo, serão utilizadas as imagens de treino e as respetivas máscaras. Nas máscaras de cada imagem serão alteradas as cores do seguinte modo: os pixéis a verde ou a preto, serão convertidos para preto e os pixéis vermelhos são convertidos para branco. Caso exista mais que uma máscara de uma determinada imagem, então é efetuado um merge dessas máscaras. A imagem 3 mostra a máscara do carro numa imagem após alteração das cores:

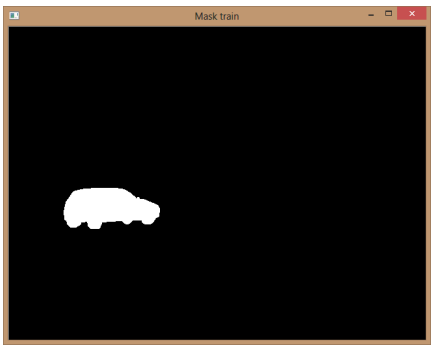

Figure 3: Máscara a preto e branco do carro.

Finalizado o tratamento das máscaras, é efetuada a deteção dos pontos característicos pelos métodos SIFT ou SURF utilizando a máscara do carro e a máscara do fundo (obtida por complemento da máscara carro). Depois é realizada a extração dos descritores Bag-of-words da imagem do carro e imagem de fundo, chamando a função compute do extrator do Bag-of-words e normalizados os descritores Bag-of-words. É gravado o ficheiro de treino em disco, com extensão YML.

Utilizamos dois classificadores: NormalBayes Classifier e SVM Classifier que são treinados com a matriz de treino e resultado criadas anteriormente, ou carregadas de ficheiro, através aos métodos train e train_auto dos classificadores NormalBayes e SVM, respetivamente.

Para o classificador SVM com train_auto só dois parâmetros foram configurados deixando os restantes para serem auto configuráveis. O tipo de SVM escolhido foi C_SVC (C-Support Vector Classification), ou seja, classificação de n classes (n igual ou maior que 2) permitindo a separação imperfeita das classes com multiplicador C de penalização para valores atípicos. O segundo parâmetro corresponde ao tipo de kernel escolhido que neste caso foi Linear, significa que não é feito o mapeamento, dis-

criminação linear (ou regressão) é feita no espaço original de características.

Na fase de teste dos classificadores, para cada imagem de teste é realizado um varrimento com janelas de scan de diferentes tamanhos (65%, 50%, 33% e 14% da janela original).

Para cada fragmento são detetados os seus pontos característicos e extraído e normalizado o descritor. Em seguida, é realizada a previsão dessa região com os dois classificadores: NormalBayes e SVM. Os dados são armazenados de dois modos, são armazenados em imagem e armazenados numa estrutura para posterior avaliação. São criados dois tipos de imagens, uma representando os fragmentos classificados como carro, e outra apenas com o fragmento classificado como carro com o valor máximo de pontos característicos. Em ambos os casos a restante região é representada a preto. Para a avaliação dos classificadores, efetuamos o seguinte procedimento, percorrer todas as dos fragmentos de scan e confrontá-las com as máscaras de treino. Se 50% ou mais pixéis desse fragmento correspondem na máscara a pixéis brancos ou se 50% ou mais do carro da máscara pertencer ao fragmento, então esse fragmento é considerado como carro, senão é considerado como não carro. Deste modo, é possível obter uma estimativa dos falsos positivos e dos falsos negativos obtidos na execução deste classificador.

3 Resultados

Na tabela 1 podem observar-se os resultados obtidos na avaliação dos classificadores utilizados no teste das imagens do dataset fornecido:

	Verdadeiros positivos	Falsos positivos		
SVM	31372 (25%)	91610 (75%)	(1)	SURF
N. Bayes	59343 (17%)	286579 (83%)	(1)	SURF
SVM	308 (43%)	400 (57%)	(2)	SURF
N. Bayes	231 (33%)	477 (67%)	(2)	SURF
SVM	32009 (27%)	87183 (73%)	(1)	SIFT
N. Bayes	58377 (18%)	275129 (82%)	(1)	SIFT
SVM	311 (44%)	397 (56%)	(2)	SIFT
N. Bayes	199 (28%)	509 (72%)	(2)	SIFT

Table 1: Resultados e comparação entre métodos e classificadores

Na tabela (1) significa que os resultados são referentes a todos os fragmentos analisados positivos e (2) significa que os resultados são referentes apenas aos fragmentos positivos com maior número de keypoints.

Nas figuras seguintes são apresentados os resultados obtidos nas deteções utilizando os algoritmos SURF e SIFT com os classificadores Normal Bayes e SVM.

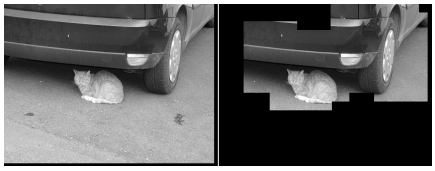

Figure 4: Área classificada como carro ao usar o detector SURF. A esquerda o classificador Normal Bayes e direita o SVM.

Figure 5: Área classificada como carro ao usar o detector SIFT. A esquerda o classificador Normal Bayes e direita o SVM.

4 Discussão dos resultados

Verificou-se que a percentagem de deteção de verdadeiros positivos é, em todos os casos, inferior a 50%, o que ficou um pouco abaixo do que era previsto. No entanto, é possível verificar que o algoritmo SIFT obtém resultados ligeiramente melhores que o algoritmo SURF, tanto para o classificador Normal Bayes como para o classificador SVM.

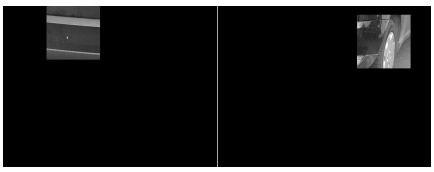

Figure 6: Janela com mais features classificada como carro ao usar o detector SURF. A esquerda o classificador Normal Bayes e direita o SVM.

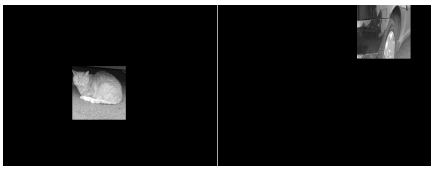

Figure 7: Janela com mais features classificada como carro ao usar o detector SIFT. A esquerda o classificador Normal Bayes e direita o SVM.

No que concerne ao tempo de execução, o sistema implementado também ficou aquém do esperado, uma vez que para a análise das cento e setenta e sete imagens do dataset fornecido o tempo de execução rondou as três horas para o método SURF enquanto que o método SIFT demorou cerca do dobro do tempo.

Relativamente à comparação entre os dois classificadores, Normal Bayes e SVM, observa-se uma vantagem considerável para o classificador SVM pois apresenta maiores percentagens de deteção de verdadeiros positivos em todos os modos de execução.

5 Conclusões e Melhoramentos

A realização deste trabalho, permitiu elaborar um programa para deteção de carros em imagens, e ainda para o aprofundamento de conhecimentos de processamento e análise de imagens e visão por computador, utilizando a biblioteca OpenCV e os algoritmos de deteção de pontos característicos SURF e SIFT.

Algumas melhorias para este projeto passariam pela redução do tempo de execução, pela utilização de outro tipo de kernel, por exemplo, do tipo Gaussian RBF e pela criação de uma interface gráfica para aumentar a usabilidade da mesma. Tal como, uma melhoria na eficácia da deteção dos carros nas imagens, que poderia ser obtida com a utilização de máscaras que melhor ajustadas ao carro na imagem.

No entanto, pensamos que o trabalho realizado apresenta qualidade e que atingiu os objetivos mínimos que se propunha.

References

[1] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, page 22, 2004.

[2] Eric Nowak, Frédéric Jurie, and Bill Triggs. Sampling strategies for bag-of-features image classification. In *Computer Vision–ECCV 2006*, pages 490–503. Springer, 2006.