

Trabalho Final de Sistemas Inteligentes

Rodrigo Kanehisa²

¹Departamento de Informática – Universidade Federal do Maranhão (UFMA)
Av. dos Portugueses, 1966 Bacanga - CEP 65080-805 – São Luís – MA – Brasil

²Universidade Federal do Maranhão (UFMA) – São Luís, MA – Brasil

rodrigokanehisa@gmail.com

Resumo. *Este relatório descreve o trabalho realizado como avaliação final para a disciplina de TECC - Visão Computacional. Estudando a aplicação de Bag of Words e SURF para detecção de faca em imagens.*

Palavras-Chave: *Detecção de Faca, Deep Learning, Bag of Words.*

1. Introdução

Circuitos Fechado de Televisão (CCTV) estão se tornando cada vez mais populares, estando presentes em quase todos os espaços públicos. Isto leva há uma enorme sobrecarga para operadores de CCTV, já que o número de câmeras que um operador consegue monitorar está limitado por fatores humanos. Um individuo portando armas em público é um forte indicador de possível situação perigosa. Estudos mostram que a resposta rápida das autoridades é o principal fator na redução do número de vítimas

2. Objetivos

- Detectar facas em imagens
- Verificar a aplicação deste sistema em situações reais
- Reduzir a ameaça de pessoas armadas

3. Fundamentação Teórica

Nesta sessão são discutidos os conceitos utilizados para o entendimento deste trabalho.

3.1. Bag of Words

O modelo de bag-of-words(BOW) é uma representação simplificada usada no processamento de linguagem natural e recuperação de informações. Neste modelo, um texto é representado como o conjunto de suas palavras, desconsiderando a gramática e mesmo a ordem das palavras, mas mantendo a multiplicidade. Representação da frequência acumulada de palavras em documentos diferentes a partir de um dicionário pré-criado.[Wikipedia 2017a]

Na visão computacional, o modelo de BOW pode ser aplicado à classificação de imagens, ao tratar as features da imagem como palavras. Na classificação de documentos, um BOW é um vetor esparsa de contagens de ocorrências de palavras. Na visão computacional, um BOW visuais é um vetor de contagens de ocorrências de um vocabulário de features locais na imagem.[Wikipedia 2017b]

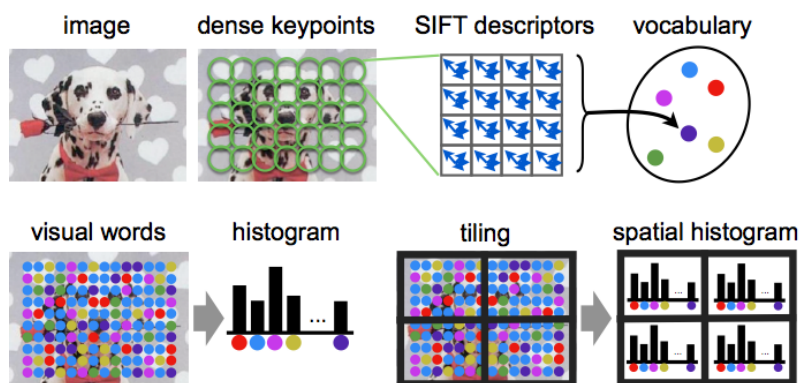


Figura 1. Exemplo do processo de classificação de imagens com BOW.

3.2. SURF

Speeded Up Robust Features ou SURF é um extrator de características de imagens. Consiste num vetor de características obtidas ao redor do keypoint localizado. É usado de forma análoga ao SIFT[Lindeberg 2012] porém ao utilizar uma aproximação com BoxFilter, ao invés do Laplaciano de Gaussiana com DoG, se obtém um ganho de em velocidade de processamento.[Bay et al. 2006]

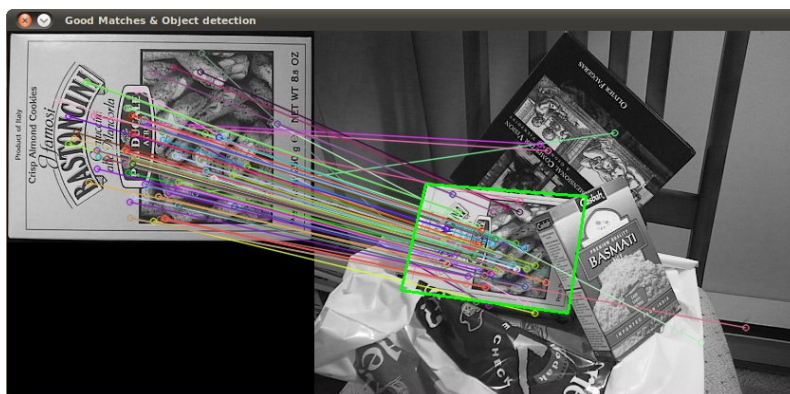


Figura 2. Resultado da aplicação do descritor SURF.

O SURF utiliza box filters para aproximação do laplaciano. A grande vantagem desta aproximação é que estas convoluções podem ser facilmente calculada com a utilização de imagens integrais. Além de poderem ser executadas em paralelo para várias escalas diferentes. Além disso, o SURF depende do determinante da matriz de hessiana tanto para a escala como para a localização.

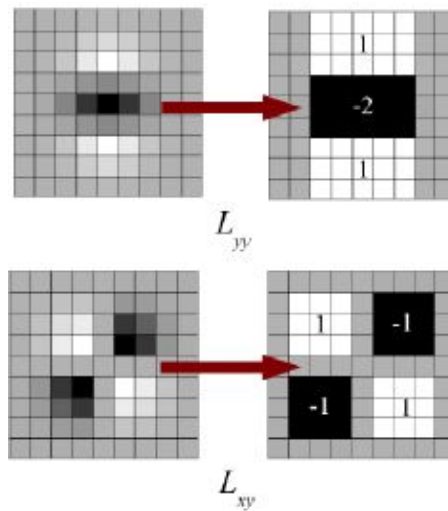


Figura 3. Aproximação do Laplaciano com Box Filter.

4. Metodologia

Nesta sessão é discutida a metodologia utilizada neste trabalho.

4.1. Base de dados

A base de dados utilizada neste trabalho foi a mesma utilizada no artigo [Grega et al. 2016]. O conjunto de dados para detecção de faca foi obtido a partir de gravações CCTV. As imagens foram cortadas dos quadros originais usando o método da janela deslizante. O tamanho da janela definido para 100 x 100 pixels cada. A base contém 9340 exemplos negativos e 3559 exemplos positivos. Os casos negativos consistem em imagens neutras, contendo todo tipo de objeto ou ambientes. Os casos positivos contém a imagem de uma faca e uma pessoa à segurando ou ao menos uma silhueta humana visível, pois uma faca só é considerada uma ameaça se estiver sendo manejada por uma pessoa.



Figura 4. Exemplo de imagens da base de dados.

4.2. Preparo da Base

Cada imagem é convertida de RGB para . Isto é feito para se executar a equalização do histograma, com o objetivo de realçar as características da imagem. Uma vez que o histograma tenha sido equalizado, a imagem é convertida de volta pra RGB. Nesta fase parte das imagens da base foram removidas já que mesmo com a equalização de histograma, não foi possível extrair características, inviabilizando a aplicação da técnica. Por fim restaram 9333 imagens negativas e 3558 imagens positivos.

Das 12891 imagens da base foram separadas 7116, contendo uma quantidade igual de casos positivos e negativos. Isto foi feito para balancear a quantidade de casos positivos e negativos. As imagens negativas utilizadas foram escolhidas de forma aleatória, já a base positiva foi usada por completo. Por fim a base foi dividida em 5044 imagens para treino e 2070 para teste.

4.3. Treinamento

A ferramenta escolhida foi o WEKA [Hall et al. 2009]. Weka é uma coleção de algoritmos de aprendizagem de máquinas para tarefas de mineração de dados. Weka contém ferramentas para pré-processamento de dados, classificação, regressão, agrupamento, regras de associação e visualização. Também é adequado para o desenvolvimento de novos esquemas de aprendizado de máquinas. Weka é um software de código aberto emitido sob a GNU General Public License.

Dentro do WEKA, foi utilizada a ferramenta Auto-WEKA [Thornton et al. 2013] para realização dos testes. Pela existência de múltiplos algoritmos de aprendizado de máquina, e pela performance dos mesmos depender muitas vezes de seus parâmetros iniciais. O Auto-WEKA permite a automatização dos testes de vários algoritmos de classificação. O testes foram executados durante um período de 8 horas.

5. Resultados

Após o período de 6 horas de testes utilizando o Auto-WEKA, o melhor classificador encontrado foi o SMO (Sequential Minimal Optimization) [Platt 1998]. Este classificador durante o treino com o Auto-WEKA obteve 91.85% de acurácia, classificando corretamente 4633 dos 5044 casos utilizados. Porém ao tentar replicar estes resultados utilizando exclusivamente a base de testes, não foram obtidos resultados parecidos.

O testes feito com a base de teste mostraram que a técnica é inefetiva para esta aplicação, acertando apenas 1314 das 2070 imagens verificadas, cerca de 63.47% de acurácia. Estes resultados mostram que os teste feito pelo Auto-WEKA durante o treino do classificador estavam em overfitting.

	Negativos	Positivos
Negativos	2280	240
Positivos	171	2353

Tabela 1. Matriz de confusão do treino do classificador

	Negativos	Positivos
Negativos	590	447
Positivos	309	724

Tabela 2. Matriz de confusão dos testes de validação

Uma hipótese para estes resultados ruins é o fato de não haver grandes diferenças entre as características extraídas pelo SURF nas imagens positivas e negativas, pois provavelmente o SURF extraía características do background das imagens positivas, estas características se assemelham muito as características obtidas nas imagens negativas impedindo o classificador de encontrar um modelo que classifique com eficiência sem cair em overfitting.

6. Conclusão

Este trabalho apresentou a avaliação de uma técnica para detecção automatizada de faca em imagens. A técnica se não se mostrou na classificação dos dados, os resultados obtidos: 61.82% de sensibilidade, 65.62% de especificidade e 63.47% de acurácia. O número alto de falsos positivos e falsos negativos inviabilizam o uso da técnica em um ambiente real. Uma possível solução seria segmentar a imagem para detecção da região em que se encontra a pessoa e subtrair o background da imagem.

Como objetivos futuros é pretendido segmentar a região contendo pessoas, ignorando o background, coisa que é inviável na base de dados atual, porém é possível em imagens reais de CCTV. Testar os métodos atuais em tempo real com vídeo. Além de utilizar descritores alternativos ao SURF como o KAZE[Alcantarilla et al. 2012] ou o ORB[Rublee et al. 2011], e descritores com outras características como Histogram of oriented gradients[Tsai 2010]. Também utilizar outras técnicas de aprendizado de máquina como deep learning, fazendo uso da ferramenta Keras[Chollet et al. 2015] utilizando o Tensor Flow[Abadi et al. 2016] como backend e a Darknet - YOLO[Redmon and Farhadi 2016].

Referências

- [Abadi et al. 2016] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I. J., Harp, A., Irving, G., Isard, M., Jia, Y., Józefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D. G., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P. A., Vanhoucke, V., Vasudevan, V., Viégas, F. B., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *CoRR*, abs/1603.04467.
- [Alcantarilla et al. 2012] Alcantarilla, P. F., Bartoli, A., and Davison, A. J. (2012). Kaze features. In *European Conference on Computer Vision*, pages 214–227. Springer.
- [Bay et al. 2006] Bay, H., Tuytelaars, T., and Van Gool, L. (2006). Surf: Speeded up robust features. *Computer vision—ECCV 2006*, pages 404–417.
- [Chollet et al. 2015] Chollet, F. et al. (2015). Keras: Deep learning for python. <https://github.com/fchollet/keras>.
- [Grega et al. 2016] Grega, M., Mاتیolański, A., Guzik, P., and Leszczuk, M. (2016). Knives images database. [Online; acessado em 19-April-2017].
- [Hall et al. 2009] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18.
- [Lindeberg 2012] Lindeberg, T. (2012). Scale invariant feature transform. *Scholarpedia*, 7(5):10491.
- [Platt 1998] Platt, J. (1998). Sequential minimal optimization: A fast algorithm for training support vector machines.
- [Redmon and Farhadi 2016] Redmon, J. and Farhadi, A. (2016). YOLO9000: better, faster, stronger. *CoRR*, abs/1612.08242.
- [Rublee et al. 2011] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). Orb: An efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE international conference on*, pages 2564–2571. IEEE.
- [Thornton et al. 2013] Thornton, C., Hutter, F., Hoos, H. H., and Leyton-Brown, K. (2013). Auto-weka: Combined selection and hyperparameter optimization of classification algorithms. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 847–855. ACM.
- [Tsai 2010] Tsai, G. (2010). Histogram of oriented gradients. *University of Michigan*.
- [Wikipedia 2017a] Wikipedia (2017a). Bag-of-words model — wikipedia, the free encyclopedia. [Online; accessed 21-July-2017].
- [Wikipedia 2017b] Wikipedia (2017b). Bag-of-words model in computer vision — wikipedia, the free encyclopedia. [Online; accessed 21-July-2017].