COMINF 4thEdition Volume1

Implementação de um modelo Bag of Features para classificação de frutas e vegetais

Guilherme H. G. Christmann[1], Ricardo B. Grando[1], Fabrício J. C. Montenegro[1], Rodrigo S. Guerra[1]

[1]Universidade Federal de Santa Maria (UFSM) - Santa Maria - RS - Brasil

Abstract. *This work explores a classic technique in computer vision, the Bag of Features (BoF) model, in a fruit and vegetable classification problem. There's an increasing trend in the use of Neural Networks and Deep Learning techniques applied to the automation of processes and systems. This work goes against this trend, examining how a simpler Machine Learning (ML) model would perform. For this, we defined two scenarios, one in a more controlled environment with differences only in light and objects positions, and another with more background clutter. We show that, although the trend is to use bigger and more complex ML models, simpler techniques continue to be relevant in certain scenarios.*

Resumo. *Este trabalho explora uma técnica clássica de visão computacional, o modelo de Bag of Features, em um problema de classificação de frutas e vegetais. Há uma tendência crescente no uso de técnicas de Redes Neurais e Deep Learning aplicadas na automatização de processos e sistemas. Esse trabalho vai contra essa tendência, examinando o desempenho de um modelo de Machine Learning (ML) mais simples. Para tal, definimos dois cenários, um em que o ambiente é controlado com apenas diferenças de iluminação e posições dos objetos e, outro, com imagens mais desafiadoras, com maior presença de ruído no background. Mostramos que apesar de existir uma clara tendência ao uso de modelos complexos de ML, técnicas mais simples continuam relevantes em certos cenários.*

1. Introdução

Grande parte dos esforços da área de *Machine Learning* (ML) atualmente são destinados à automatização de processos e serviços previamente realizados por humanos. O aumento da capacidade de processamento nos últimos anos proporcionou o barateamento de CPUs e GPUs poderosas, de forma a estabelecer um ambiente propício para o desenvolvimento e aprimoramento das técnicas de ML. A aplicação desses modelos, que muitas vezes supera o desempenho de humanos, cada vez mais faz parte da rotina das pessoas e tem-se provado uma ferramenta valiosa para profissionais de diversas áreas. Um processo que pode fazer uso da automatização é a classificação para pesagem de frutas e vegetais em supermercados, que ainda é feita por funcionários humanos, geralmente consultando uma tabela visual para conferir o código do produto e inserindo o mesmo em uma balança para determinar o preço. Um dos impedimentos da implementação de sistemas automatizados em supermercados é o custo de desenvolvimento de tais sistemas, incluindo o *hardware* necessário para o processo.

Técnicas de classificação e detecção têm-se tornado cada vez mais precisas, com uma preferência crescente ao uso de Redes Neurais (RNs) e *Deep Learning*

[LeCun et al. 2015]. Porém, RNs são computacionalmente custosas de serem treinadas, demandando horas de processamento, além de necessitarem de uma grande quantidade de amostras para atingir um desempenho robusto. Outro fator que deve ser levado em consideração é o fato de que RNs, geralmente, dependem fortemente de um *hardware* de alta performance para realizar suas predições em tempo real, tornando difícil justificar seu uso em sistemas pequenos e embarcados, como no cenário apresentado nesse trabalho.

Em classificação de frutas e vegetais, ótimos resultados foram obtidos utilizando fusão de diferentes *features* após extração do *background* em imagens de frutas [Rocha et al. 2010]. Em [Zhang et al. 2014] é utilizada uma técnica semelhante onde PCA [Ke and Sukthankar 2004] é aplicado à diferentes *features* para reduzir sua dimensionalidade e posteriormente classificadas por uma RN, alcançando uma precisão de classificação de 89,1%. Operando diretamente sobre as imagens com *background* extraído, [Zhang et al. 2017] utilizaram uma RN de convolução (CNN) profunda com 13 camadas e técnicas de aumento de dados [Perez and Wang 2017], alcançando uma precisão de 94,94%, porém uma precisão menor quando aplicado nas imagens com a prsença de *background*. No entanto, técnicas de remoção automática de *background* necessitam de um certo controle do ambiente, podendo falhar em imagens que contenham muito *background clutter*. [Akbari Fard et al. 2016] demonstra que CNNs simples com menos camadas não atingiram bons resultados em um *dataset* com ambiente menos controlado e sem extração de *background*, conseguindo apenas 45% de precisão.

Este artigo explora um método de classificação de frutas e vegetais em imagens utilizando técnicas clássicas de *Machine Learning*, comparando seu desempenho em dois *datasets* distintos, o mesmo de [Rocha et al. 2010] com algumas modificações e outro de nossa própria autoria contendo *background clutter*. O método utilizado para geração de *features* é conhecido como *Bag-of-Features* (BoF) ou *Bag-of-Visual-Words* [Csurka et al. 2004], uma adaptação do método de análise textual chamado *Bag-of-Words* (BoW) para o contexto de visão computacional. Em [Anthimopoulos et al. 2014], o método de BoF foi utilizado em um cenário de detecção de comida para diabéticos, atingindo bons resultados com detecção de *features* SIFT, agrupamento por *K-means* e um classificador SVM. Recentemente também foi aplicado em parte na classificação de imagens de ressonância magnética para determinar a ocorrência de lesões cerebrais [Minaee et al. 2017].

2. Metodologia

O BoF é um método em visão computacional com raízes em uma técnica clássica para classificação de texto conhecida como *Bag of Words* (BoW) [Sebastiani 2002]. No BoW a classificação é feita a partir de uma representação alternativa do texto na forma de um histograma. Para gerar essa representação alternativa é necessária uma etapa prévia de agrupamento (*clustering*) [Caruana and Niculescu-Mizil 2006] para construir o dicionário de palavras que serão procuradas no texto. A representação de histograma, então, é gerada contando o número de ocorrências de cada palavra do dicionário dentro do texto a ser classificado.

No contexto de visão computacional, o método possui forma similar, porém o agrupamento é feito a partir de *features* retiradas de imagens, ao invés de texto. Há diversas técnicas para determinar regiões de imagem relevantes para a extração de *fea-*

tures [Rui et al. 1999], como algoritmos de detecção de cantos [Awrangjeb et al. 2012], simplesmente definindo regiões a cada N *pixels* ou até definindo regiões aleatoriamente. Em nosso método optamos pelo algoritmo SIFT (*Scale Invariant Feature Transform*) [Lowe 1999] por apresentar robustez à variações de escala, orientação e diferenças de iluminação. Após definidas as regiões ou *keypoints* através do SIFT, é aplicada a etapa de extração de *features*, onde, para cada *keypoint*, é calculado um descritor SIFT. Um descritor SIFT é um vetor de 128 números de ponto flutuante, que representa a identidade visual do *keypoint* na imagem. O descritor é calculado aplicando HoG [Dalal and Triggs 2005] em uma região 16x16 na vizinhança do *keypoint*. Os descritores SIFT, então, são as *features* utilizadas para a construção do dicionário, e posteriormente, para classificação em uma imagem nova.

Para construir nosso dicionário visual utilizamos a técnica de agrupamento *K-means* [Duda et al. 2012]. Um vetor de features e um parâmetro K, que define o número de agrupamentos a ser realizado, são fornecidos ao algoritmo. Através de um treinamento não supervisionado, descritores/*features* semelhantes são colocados no mesmo grupo.

A partir disso, é possível gerar uma representação de histograma de uma nova imagem. Gera-se seus *keypoints* e descritores SIFT e, para cada um deles, confere-se em qual grupo ou "palavra visual" do dicionário o descritor melhor se assemelha, incrementado uma ocorrência para aquele grupo. Ao final do processo tem-se uma distribuição da frequência de ocorrência de cada palavra, um histograma. Na Figura 1 é apresentado uma imagem com os *keypoints* SIFT e o histograma gerado a partir de seus descritores.

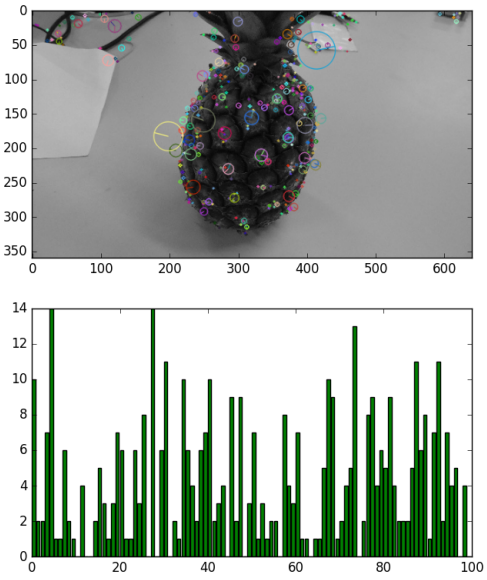

Figura 1. Histograma com dicionário de 100 palavras

Os passos para realização do método são os seguintes:

1. Extração de vetores de *features* utilizando SIFT.
2. Agrupamento dos vetores de *features* através de *K-means*, formando as palavras visuais.

3. Representação das imagens através de histogramas dessas palavras visuais.
4. Treinamento de um classificador SVM utilizando os histogramas.

Para medir o desempenho do método utilizamos dois *datasets*: (1) um de nossa própria autoria, com 3 categorias de frutas, além de uma categoria de *background*, e (2) uma versão modificada do *dataset* de frutas e vegetais da Unicamp [Rocha 2010].

2.1. *Dataset* modificado da Unicamp

O *dataset* da Unicamp [Rocha 2010] apresenta imagens em um ambiente controlado, com fotos das frutas e vegetais em cima de uma mesa, com variações de iluminação e posição dos elementos. Originalmente o *dataset* continha 15 categorias de frutas e vegetais, porém, removemos a categoria *onion* (cebola) por julgarmos conter imagens insuficientes para a aplicação do método (apenas 75). Além disso, removemos algumas imagens que estavam muito desfocadas, de forma a prejudicar a detecção dos *keypoints* SIFT. Apesar dessas modificações, julgamos que o *dataset* permanece representativo para o cenário proposto. O *dataset* final contém um total de 2463 imagens nas seguintes categorias:

1. *agata_potato* (199);
2. *asterix_potato* (180);
3. *cashew* (209);
4. *diamond_peach* (208);
5. *fuji_apple* (208);
6. *granny_smith_apple* (152);
7. *honneydew_melon* (125);
8. *kiwi* (159);
9. *nectarine* (242);
10. *orange* (102);
11. *plum* (252);
12. *spanish_pear* (136);
13. *taiti_lime* (104);
14. *watermelon* (187).

Figura 2. Exemplos *Dataset* da Unicamp

Para construir o dicionário, 40% das imagens de cada categoria foram selecionadas aleatoriamente, totalizando 978 imagens. O conjunto restante foi dividido em treino e validação. Para treino foram retiradas aleatoriamente 55 imagens de cada categoria, totalizando 770 imagens. O restante das imagens de cada categoria foram utilizadas para validação, com um total de 715 imagens. Na Figura 2 são apresentados alguns exemplos de imagens neste *dataset*.

2.2. *Dataset* Próprio

Para testar o desempenho do método em cenários mais desafiadores, onde ocorre uma presença maior de *background clutter* optamos por construir nosso próprio *dataset*. Definimos 3 categorias de frutas (Abacaxi, Melão e Banana), além de uma categoria de *background*. Para tal, utilizamos uma câmera da Sony, modelo DSC-H100 com sensor CCD de 16 MP. Dividimos a construção do nosso *dataset* em duas etapas, uma para construção do dicionário e outra para treinar e validar o classificador.

2.2.1. *Dataset* para construção do dicionário

Fotografamos as 3 frutas em um ambiente controlado, com painéis brancos ao fundo, em diversos ângulos e orientações. Posteriormente, removemos o fundo das imagens de forma a restar apenas a informações das frutas na mesma, visando reduzir o número de *features* indesejadas no processo de *clustering*. Para a categoria de *background*, fotografamos diversas áreas e objetos de nosso laboratório. Na Figura 3 são apresentados exemplos dessas imagens. O *dataset* para construção do dicionário totalizou 331 imagens com as seguintes categorias:

1. *background* (87);
2. *banana* (136);
3. *melon* (39);
4. *pineapple* (69);

Figura 3. Exemplos *Dataset* para construção do Dicionário

2.2.2. *Dataset* para Classificador

Para criar o *dataset* para treinar o classificador fotografamos as 3 frutas em diversas áreas de nosso laboratório, além de fotografar a mesma área sem nenhuma fruta. A motivação é de que o classificador aprenda a diferenciar uma imagem que contém apenas *background* de uma imagem que contém uma fruta com presença de *background*. Na Figura 4 apresentamos alguns exemplos dessas imagens. Este *dataset* possui um total de 215 fotos:

1. *background* (51);
2. *banana* (55);
3. *melon* (55);
4. *pineapple* (54);

Figura 4. Exemplos *Dataset* para Classificador

Como nosso *dataset* para o classificador possui um número pequeno de amostras, aplicamos 5 rodadas de validação cruzada a fim de estimar sua precisão de forma mais confiável. Em cada rodada, retiramos aleatoriamente 45 imagens de cada categoria para treino, e o restante para validação. Dessa maneira, temos 180 imagens para treino e 35 imagens para validação em cada rodada.

2.3. Construção do Dicionário Visual

Para cada uma das categorias são computados os descritores SIFT de todas as imagens. Para manter um balanceamento, as *features* são ordenadas de forma decrescente a partir do parâmetro *response*, que representa a qualidade do *keypoint* de acordo com o algoritmo SIFT. Valores maiores indicam uma maior probabilidade de que o *keypoint* seja reencontrado em outras imagens do objeto. A quantidade de *features* a ser selecionada de cada categoria é definida como 80% do número de *features* da categoria com o menor número. No *dataset* 2.2.2, *Banana* foi a categoria com menor número, 39069, de forma a serem selecionadas as 31255 melhores *features* de cada categoria, totalizando 125020 *features* para realizar o agrupamento. Para o *dataset* 2.1, a categoria com menor número de *features* foi *honneydew_melon* com 2719. Selecionamos, então, 2175 *features* de cada categoria, totalizando 30450 para realizar o agrupamento.

Existem técnicas para estimar o número ótimo de grupos em que um determinado conjunto de dados deve ser separado. Porém, devido à natureza do nosso método, onde o classificador opera em cima de uma representação de histogramas, o número de grupos dentro do dicionário acaba não tendo uma influência tão grande em sua precisão. Apenas é necessário que hajam grupos suficientes para gerar uma representação de histogramas distintas, de acordo com o número de categorias a serem classificadas. Na seção de resultados é possível ver que o efeito do número de grupos foi maior no *dataset* 2.1 com 14 categorias, comparado ao *dataset* 2.2.2 com apenas 4 categorias. Geramos, então, 10 agrupamentos diferentes, de 100 a 1000 grupos, com incrementos de 100.

2.4. Treinamento do Classificador SVM

Com os agrupamentos realizados podemos representar as imagens através de histogramas. Para treinar o classificador SVM geramos os histogramas de todas as imagens dentro do *dataset* de treino. Posteriormente, fornecemos à SVM o histograma de cada imagem junto com um rótulo da categoria à qual aquele histograma pertence (treinamento supervisionado). A SVM é sensível ao balanço de dados por categoria na etapa de treinamento, portanto fornecemos o mesmo número de histogramas (45) de cada categoria. Além disso, também é aplicada uma etapa de normalização dos histogramas, dividindo cada ocorrência em uma *palavra* do dicionário pelo número total de ocorrências. Isso é feito para que o número de *features* retiradas de uma imagem não influencie sua classificação, apenas o formato de seu histograma.

Quando o modelo da SVM é criado existem diversos hiper-parâmetros que ficam a cargo do desenvolvedor definir. Esses parâmetros afetam diretamente o desempenho do classificador, de forma que é necessária realizar uma busca para definir os melhores parâmetros. Uma SVM possui os parâmetros de *kernel* (podendo ser linear ou *RBF*), um parâmetro de regularização **C** e um valor de tolerância γ. Para definir qual o melhor conjunto de parâmetros para cada dicionário é aplicada uma busca extensiva para cada conjunto possível, buscando a maior precisão. Para **C**, valores de 0 a 1000 com incrementos de 100, e para γ, os valores 0.1, 0.01, 0.001 e 0.0001.

3. Resultados

Esta seção foi dividida em duas subseções, apresentando os resultados da aplicação do método nos dois *datasets* discutidos anteriormente, comparando tamanho de dicionário com precisão de classificação. Nas Tabelas 1 e 2 também são apresentadas algumas métricas por categoria. **Precisão** é definida como $tp/(tp + fp)$, onde tp é o número de positivos verdadeiros e fp o número de falsos positivos. ***Recall*** é a razão $tp/(tp + fn)$ onde tp é o número de positivos verdadeiros e fn o número de falsos negativos. **Suporte** é o número de ocorrências de cada classe.

3.1. *Dataset* Unicamp

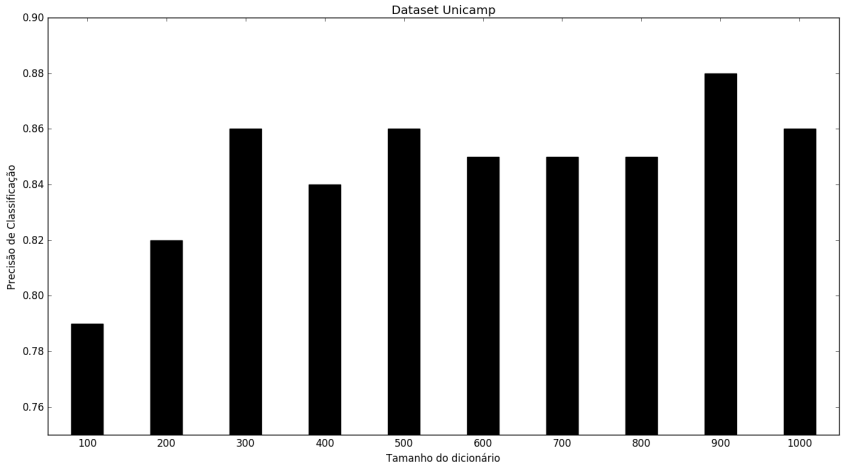

Figura 5. Comparação de tamanho do dicionário com precisão de classificação no *dataset* da Unicamp.

Na Figura 5 podemos ver a relação entre o número de palavras do dicionário visual e seu efeito na precisão de classificação. A melhor precisão de classificação foi de 88%, com um dicionário de 900 palavras visuais. Os parâmetros utilizados para o classificador SVM foram *kernel* do tipo **RBF**, **C** igual a 100 e γ igual a 0.1. Na Tabela 1 apresentamos a média de desempenho em cada categoria do *dataset*.

3.2. *Dataset* Próprio

Como o *dataset* de nossa criação possui um pequeno número de amostras, aplicamos 5 rodadas de validação cruzada. Os resultados apresentados na Figura 6 são a média das 5 rodadas, para cada tamanho de dicionário.

Categoria	Precisão	*Recall*	Suporte
agata_potato	0.85	0.69	65
asterix_potato	0.60	0.75	53
cashew	1.00	1.00	71
diamond_peach	0.82	0.87	70
fuji_apple	0.91	0.87	70
granny_smith_apple	0.91	0.86	37
honneydew_melon	0.47	0.90	20
kiwi	0.83	0.83	41
nectarine	0.95	0.79	91
orange	1.00	1.00	7
plum	0.95	0.86	97
spanish_pear	0.65	0.81	27
taiti_lime	1.00	1.00	8
watermelon	1.00	0.98	58
Média/Total	0.88	0.85	715

Tabela 1. Métricas por categoria - *Dataset* Unicamp

A melhor precisão obtida foi utilizando o dicionário que continha 700 palavras (média de 0.808, comparado ao de 900 com 0.802). Na Tabela 2 apresentamos as métricas médias da validação cruzada, por categoria.

Categoria	Precisão	*Recall*	Suporte
background	0.422	0.766	6
banana	0.798	0.5	10
melon	0.934	0.84	10
pineapple	0.936	0.89	9
Média/Total	0.808	0.742	35

Tabela 2. Métricas por categoria - *Dataset* Próprio

4. Discussão

Observando os resultados em ambos os *datasets* pode-se notar a influência do tamanho do dicionário no desempenho do classificador. Para o *dataset* da Unicamp a variação de precisão foi maior, aumentando de 76% com 100 palavras, para 88% com 900 palavras. Já em nosso *dataset in-house* a variabilidade foi menor. Uma explicação possível para esse fenômeno é a diferença no número de categorias dos *datasets*. Um dicionário maior permite que um maior número de *features* ocupe palavras diferentes, gerando uma representação de histograma mais diferenciada, quando comparada com representações a partir de dicionários menores.

É interessante notar que, apesar de termos obtido uma precisão acima de 80%, há uma grande variabilidade no desempenho de classificação entre as categorias. Isso pode ser explicado pela natureza das imagens que estamos classificando. Nosso método não leva em consideração informações de cor, apenas os vetores de descritores SIFT e sem remoção de *background*. Dessa maneira, frutas ou vegetais que possuem textura parecida porém cores diferentes acabam sendo classificadas na mesma categoria.

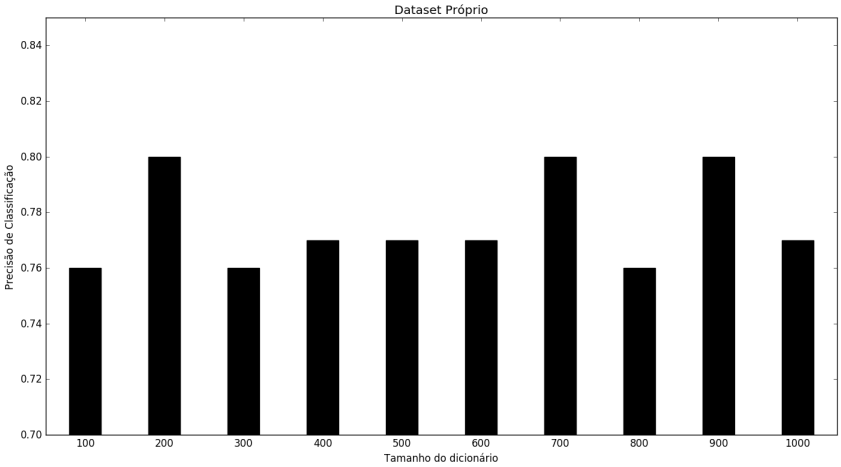

Figura 6. Comparação de tamanho do dicionário com precisão de classificação em nosso *dataset*.

CNN com 3 camadas [Akbari Fard et al. 2016]	**CNN com 13 camadas** [Zhang et al. 2017]	**BoF - SIFT** (Nosso)
45.00%	89.60%	80.80%

Tabela 3. Comparação de precisão com outros modelos em imagens com presença *background clutter*

Observando os resultados em nosso *dataset in-house* e Tabela 3 conseguimos notar que o método se manteve robusto em imagens com presença de *background*, conseguindo uma precisão bem melhor que a RN de convolução com poucas camadas usada em [Akbari Fard et al. 2016] e um pouco pior que a RN de 13 camadas em [Zhang et al. 2017]. Levando em consideração que nosso *dataset in-house* possui imagens mais desafiadoras que as de ambiente controlado utilizado em [Zhang et al. 2017], isso prova que métodos simples podem ser comparáveis à RNs grandes nesse contexto, sem a necessidade de alavancar o uso de GPUs para atingir uma boa performance.

Apesar da precisão classificando imagens que continham apenas *background* não ter sido boa (42%) o classificador conseguiu distinguir as imagens que continham frutas e *background*. Um possível método para resolver tanto esse problema quanto o da variabilidade entre as categorias é utilizar *features* que contém informações de cor, juntamente com as *features* SIFT.

5. Conclusão

Apesar de existir uma clara tendência ao uso de Redes Neurais e *Deep Learning* em problemas de classificações de imagens, demonstramos que o método clássico de *Bag of Features* continua sendo relevante. O método se mostrou eficaz, utilizando de *datasets* relativamente pequenos e um baixo custo computacional de treinamento, quando comparados à RNs. Conseguimos obter uma precisão de 88% no *dataset* da Unicamp que apresenta frutas e vegetais em um ambiente controlado. Além disso, também experimentamos o método em um *dataset* de nossa construção, com imagens mais desafiadoras, que apresentam uma quantidade considerável de *background clutter*. Nesse *dataset* a melhor

precisão obtida foi de 80%, provando que o método possui potencial mesmo em ambientes não-controlados e melhor que modelos de RNs de convolução com poucas camadas. Ainda há diversas variações no método que podem ser exploradas, como utilizar diferentes algoritmos de extração de *features*, incluindo *features* de cor e fusão de *features*. Além disso, é interessante realizar experimentos para determinar se o método escala bem para um número maior de categorias.

Referências

Akbari Fard, M., Hadadi, H., and Tavakoli Targhi, A. (2016). Fruits and vegetables calorie counter using convolutional neural networks. In *Proceedings of the 6th International Conference on Digital Health Conference*, pages 121–122. ACM.

Anthimopoulos, M., Gianola, L., Scarnato, L., Diem, P., and Mougiakakou, S. G. (2014). A food recognition system for diabetic patients based on an optimized bag-of-features model. *IEEE J. Biomedical and Health Informatics*, 18(4):1261–1271.

Awrangjeb, M., Lu, G., and Fraser, C. S. (2012). Performance comparisons of contour-based corner detectors. *IEEE Transactions on Image Processing*, 21(9):4167–4179.

Caruana, R. and Niculescu-Mizil, A. (2006). An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning*, pages 161–168. ACM.

Csurka, G., Dance, C., Fan, L., Willamowski, J., and Bray, C. (2004). Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1–2. Prague.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE.

Duda, R. O., Hart, P. E., and Stork, D. G. (2012). *Pattern classification*. John Wiley & Sons.

Ke, Y. and Sukthankar, R. (2004). Pca-sift: A more distinctive representation for local image descriptors. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–II. IEEE.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee.

Minaee, S., Wang, S., Wang, Y., Chung, S., Wang, X., Fieremans, E., Flanagan, S., Rath, J., and Lui, Y. W. (2017). Identifying mild traumatic brain injury patients from MR images using bag of visual words. *CoRR*, abs/1710.06824.

Perez, L. and Wang, J. (2017). The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*.

Rocha, A., Hauagge, D. C., Wainer, J., and Goldenstein, S. (2010). Automatic fruit and vegetable classification from images. *Computers and Electronics in Agriculture*, 70(1):96–104.

Rocha, Anderson; Hauagge, D. C. W. J. G. S. (2010). Fruits/vegetables image data set collected on our local fruits and vegetables distribution center (ceasa). `http://www.ic.unicamp.br/~rocha/pub/downloads/tropical-fruits-DB-1024x768.tar.gz`. Accessed: 2017-11-21.

Rui, Y., Huang, T. S., and Chang, S.-F. (1999). Image retrieval: Current techniques, promising directions, and open issues. *Journal of visual communication and image representation*, 10(1):39–62.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47.

Zhang, Y., Wang, S., Ji, G., and Phillips, P. (2014). Fruit classification using computer vision and feedforward neural network. *Journal of Food Engineering*, 143:167 – 177.

Zhang, Y.-D., Dong, Z., Chen, X., Jia, W., Du, S., Muhammad, K., and Wang, S.-H. (2017). Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation. *Multimedia Tools and Applications*, pages 1–20.