Methods of learning discriminative features for automated visual inspection

Matthias Richter

Vision and Fusion Laboratory
Institute for Anthropomatics
Karlsruhe Institute of Technology (KIT), Germany
matthias.richter@kit.edu

Technical Report IES-2014-09

Abstract: At the present day, automation of visual inspection tasks is a typical engineering problem. Experts design the physical aspects of the system and devise classification algorithms based on a small sample of the material to be inspected. Much of this work is devoted to finding suitable features to discriminate wanted from unwanted material. In this report, we explore methods to automatically *learn* object descriptors from a suitably large sample. We focus on two types of descriptors: (a) global descriptors, which represent the object as a whole and (b) local descriptors, which focus on topical features. Apart from freeing the engineers to attend to other tasks, these methods allow non-experts to operate and reuse visual inspection systems, e.g. to inspect a different product than originally intended.

1 Introduction

Automated visual inspection is becoming more and more prevalent in many industries, from detection of precious ores and minerals in mining to quality inspection of food. Novel, improved sensors make even more application areas accessible, and the increasing speed of visual inspection systems allows previously unseen throughputs. That development as well as decreasing costs of the hardware components raises the demand even further.

One might think that it would be possible to acquire complete off-the-shelve solutions. While that is true to some degree, adapting existing systems to specific products still remains a complicated endeavour. In many cases there are best practices to approach the design of the physical aspects of the machine (i.e. material transport, lighting and image acquisition, etc.) that require moderate effort to adapt to a given problem. The major engineering bottleneck resides in establishing the classification stage. An expert identifies discriminative features by analyzing a relatively small sample of wanted and unwanted material and then devises algorithms to extract these features from an image. This lengthy process is typically driven by trial-and-error and requires extensive experience on part of the expert.

There are approaches to "widen" this bottleneck by utilizing machine learning techniques. Here, low level-features such as hue-histograms, Gabor filter responses and Fourier shape descriptors are extracted from the images and fed into generic classification algorithms like support vector machines, random forests and artificial neural networks. For encompassing reviews of these methods, see the works of Malamas et al. [MPZ+03], who investigate these approaches in visual inspection in general, and Du and Sun [DS06], who focus their attention to the application to food in particular.

However, these methods are rarely applied in practice. The main reasons are that they require far too much time to compute the features and that the black box nature of the machine learning algorithms prevents users to understand how the visual inspection system derives a decision [BCGS+09].

Additionally, we argue that such an approach misses the original question: What features are characteristic of the material and relevant to the visual inspection problem? In this report, we try to answer this question by answering another: How can one learn discriminative feature transformations from a suitably large sample of wanted and unwanted material?

1.1 Related Work

There seem to be surprisingly few researchers also concerned with this topic. In an early work, Duffy et al. propose to detect burn marks on filters by recording color histograms of images containing defects and images that show intact filters [DCL00]. These histograms are fused into a "true target' histogram that characterizes the burn marks. Defects are then detected by back-projection and thresholding the resulting image. In a follow-up study, Bergasa, Duffy et al. use the same approach, but model only non-defective color using a Gaussian mixture model of the joint red and green color distribution [BDLM00].

Zhang et al. use a similar method to grade date maturity [ZLTL14]. In training, they collect the RG-color histograms of date samples of different maturity grades and compute a back-projection table that maps RG-tuples to a ripeness level. In this table, missing values are filled in by linear interpolation between the neighboring entries, so that every possible color is represented. In testing, fruits are graded by determining the mean gray value in the back-projected image.

Similarly, Li et al. grade the ripeness of tomato fruits by determining dominant colors in images of fruits of different maturity grades [LCG09]. Their method is relatively involved and requires multiple color space conversions and a specialized clustering algorithm. Finally, they define several ripeness classes and compute characteristic histograms of dominant colors in these classes. Unseen tomatoes are classified by comparing the histogram of dominant colors to the ones learned in the training phase.

While all these approaches are concerned with a particular application in mind, Richter and Beyerer propose a method suitable for a wider range of products [RB14]. Similar to Duffy et al. they collect color histograms of the material under inspection and compute a mapping to a semantic attribute in a four step process. Unlike the other approaches, their approach requires a separate classifier that uses the attribute-images to classify objects.

All these methods focus on color as sorting criterion. Therefore, defects that are characterized by a change of texture or shape are undetectable. In the next section, we propose two methods that are able to capture all aspects of object appearance: color, texture and shape.

2 Methods

Feature descriptors can be divided into two groups: global descriptors and local descriptors. The former summarize the object appearance: “The apple is green with a few brown spots”. The latter focus on topical features: “There is a small round brown spot near the stem of the apple”. Both types of descriptors are applicable in different scenarios. Judging the overall appearance of an object calls for usage of global descriptors, e.g. when grading the surface of tiles or when assessing the ripeness of a fruit. Local descriptors, on the other hand, are applicable when searching for localized defects, for example detection of fungal infection on grains or localization of scratches on surfaces.

In this section, we present approaches to learn either type of descriptor. The first method is based on the bag of visual words (BOV) framework, while the latter utilizes cascades of random ferns.

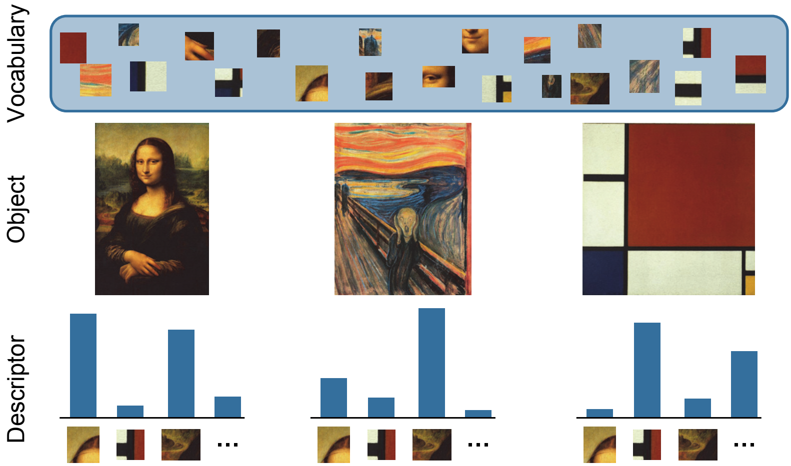

Figure 2.1: Outline of the BOV method: Given a vocabulary of visual words, objects (images) can be characterized by a statistic of the visual words appearing in the image. Note that spatial information is not considered.

2.1 Global Descriptors

Originally introduced by Csurka et al. in the context of image categorization [CDF+04], the bag of visual words model has been applied in many domains such as content based retrieval, face recognition and action classification. The descriptors have many advantages: they are compact, invariant to object scale and rotation and are highly discriminative.

The main idea is to consider images to be documents that are composed of *visual words*. As with text documents, some of the words carry more information than others, that is some words are characteristic of certain objects or concepts, while others may be found in many different images. The task of image categorization can then be approached by constructing a dictionary of discriminative keywords and describing images by determining which words appear in them. This idea is outlined in Figure 2.1. It is formalized in the following.

2.1.1 Vocabulary

Starting from a collection of input images $\mathcal{I}_i$, $i = 1, \dots, N$, a large set of local low level, D-dimensional feature descriptors $\mathbf{x}_{ti}$, $t = 1, \dots, T_i$ (e.g. SIFT) are extracted from each image. Here, T_i denotes the number of features extracted from

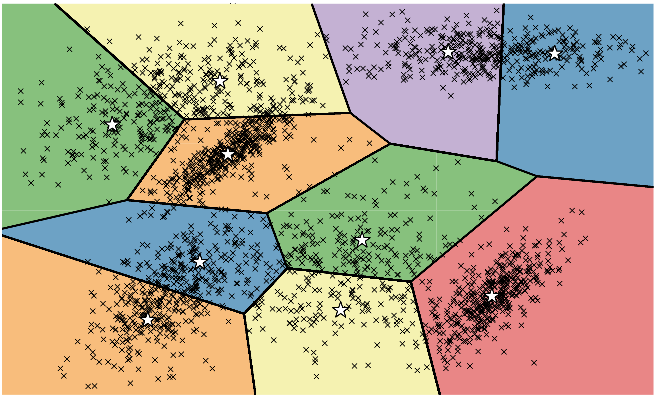

Figure 2.2: Example of hard assignment in two dimensions. Each feature (crosses) is related to the closest of the ten visual words (stars).

$\mathcal{I}_i$ and may vary from image to image. Each $\mathbf{x}_{ti}$ can be interpreted as an "inflection" of a visual word. Visual words correspond to clusters in the feature space. Therefore, keywords can be determined using cluster-analysis. Although other approaches are conceivable, most BOV implementations use a simple K-means clustering or Gaussian mixture models (GMM) with a fixed number of clusters. Csurka et al. noted that the exact number of visual words does have a negligible impact on the classification performance [CDF+04]. A common approach is to repeat the cluster analysis with varying K and keep the clustering that best fits the data.

2.1.2 Descriptors

Now that the vocabulary is determined, the descriptor $\mathbf{m}$ for an image $\mathcal{I}$ is chosen to represent some statistic over the visual words that appear in $\mathcal{I}$. Csurka et al. proposed hard assignment [CDF+04]: The D-dimensional low level features $\mathbf{x}_t$, extracted from $\mathcal{I}$, are assigned to the nearest cluster center (see Fig. 2.2). Each entry m_k in the descriptor $\mathbf{m}$ represents the fraction of features falling into the Voronoi-region around the cluster center $\boldsymbol{\mu}_k$,

$$m_k = \frac{1}{T}\left|\left\{\mathbf{x}_t \,\middle|\, \arg\min_{\boldsymbol{\mu}} \|\mathbf{x}_t - \boldsymbol{\mu}\| = \boldsymbol{\mu}_k\right\}\right|.$$

The resulting K-dimensional descriptor represents a simple count statistic of visual words, but does not provide any further information over the "inflection" of the words, i.e. the location of the features in relation to the cluster center. Fisher vectors, introduced by Perronnin et al. [PD07], provide an alternative encoding that enriches the descriptor by first-order statistics of the feature distribution.

The key idea is to assume that the $\mathbf{x}_{ti}$ are generated by a Gaussian mixture,

$$p(\mathbf{x}|\boldsymbol{\lambda}) = \sum_{k=1}^{K} \omega_k g(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k),$$

where $\boldsymbol{\lambda} = (\omega_k, \boldsymbol{\mu}_k, \Sigma_k)_{k=1}^{K}$ contains the GMM's parameters and $g(\mathbf{x}|\boldsymbol{\mu}, \Sigma)$ is a Gaussian with mean $\boldsymbol{\mu}$ and diagonal covariance matrix Σ. The parameters of the GMM are determined using expectation maximisation. Using the occurrence probabilities

$$\gamma_{kt} = \frac{\omega_k g(\mathbf{x}_t|\boldsymbol{\mu}_k, \Sigma_k)}{\sum_{j=1}^{K} \omega_j g(\mathbf{x}_t|\boldsymbol{\mu}_j, \Sigma_j)},$$

the $(2KD)$-dimensional descriptor (D is the dimension of the feature space) is built as $\mathbf{m} = \left(\mathbf{u}_1^\top, \ldots, \mathbf{u}_K^\top, \mathbf{v}_1^\top, \ldots, \mathbf{v}_K^\top\right)^\top$, where

$$\begin{aligned}
\mathbf{u}_k &= \frac{1}{N\sqrt{\omega_k}} \sum_{t=1}^{T} \gamma_{kt} \Sigma_k^{-\frac{1}{2}} (\mathbf{x}_t - \boldsymbol{\mu}_k), \text{and} \\
\mathbf{v}_k &= \frac{1}{N\sqrt{2\omega_k}} \sum_{t=1}^{T} \gamma_{kt} \left[(\mathbf{x}_t - \boldsymbol{\mu}_k)^\top \Sigma_k^{-\frac{1}{2}} (\mathbf{x}_t - \boldsymbol{\mu}_k) - 1 \right].
\end{aligned}$$

2.1.3 Application in Visual Inspection

The fundamental requirements of automated visual inspection are quite different from other computer vision tasks such as image categorization. In some sense it is less difficult, since the environmental conditions – lighting, background, etc. – are tightly controlled. On the other hand, it is more difficult than other tasks: high throughput demands very short processing times. This prohibits usage of long processing pipelines and computation of complex feature descriptors such as SIFT. This suggests to divert from the usual path in two aspects: Dense sampling and usage of primitive feature descriptors.

In **dense sampling**, we consider every foreground-pixel (u, v) of an object as key-point where to extract the low level local feature descriptors. This has the benefit of skipping an interest point detection stage (thereby saving processing time), but is only feasible if the objects are relatively small and the descriptors themselves are inexpensive to compute. Hence, we use **primitive features** that require only very little processing time. In particular, the most basic descriptor is the color of a pixel, $\mathbf{x}_t = \mathcal{I}(u_t, v_t)$. Since K-means and GMM, like many other algorithms for cluster analysis, rely on measuring distances, the color space may have a significant impact on the discriminative ability of the object descriptor. To include other features, a D-channel local descriptor is constructed as $\mathbf{x}_t = (x_{1t}, \dots, x_{Dt})^\top$, where each x_{dt} encodes a different feature-type. For example, x_{1t} to x_{3t} could correspond to the RGB values at the pixel (u_t, v_t). Texture is encoded e.g. using gradient magnitude, $x_{dt} = |\nabla\mathcal{I}(u_t, v_t)|$, rotation invariant uniform local binary patterns [OPM02] or center-symmetric local binary patterns [HPS06]. Finally, the shape of an object can be represented using the distance transform.

2.2 Local Descriptors

As BOV describes the object appearance as a whole, it does not provide information about the location of a defect. As a result, the method is unsuitable in situations where position is the major discriminative feature. In this section, we propose a novel method to learn descriptors that are able to describe such situations.

Similar to BOV, the descriptor is a collection of local features. However, instead of vectorial feature descriptors, we use very simple and *weak* binary features

$$f_n = \mathbf{1}\left[\phi_{n,1}(\mathbf{p}_{n,1}) - \phi_{n,2}(\mathbf{p}_{n,2}) < \tau_n\right], \tag{2.1}$$

where τ_n is a feature-specific threshold, $\phi(\mathbf{p})$ extracts some scalar features at the pixel at $\mathbf{p} = (u, v)$ and $\mathbf{1}[x]$ is the indicator function that is 1 if x is true and 0 otherwise. Note that neither necessarily $\mathbf{p}_{n,1} \neq \mathbf{p}_{n,2}$, nor $\phi_{n,1} = \phi_{n,2}$, which allows local features that evaluate the same pixel and features that compare different types of features.

The scalar features $\phi(\mathbf{p})$ are used to encode different aspects of the object and may be arbitrarily constructed. The only requirement is that the difference $(\phi(\mathbf{p}) - \phi(\mathbf{q}))$ is meaningful, that is ϕ should not extract some coding-scheme like LBP. However, as computation time is limited, simple extraction methods similar as the ones used in the BOV approach should be used. The hue or saturation at the location $\mathbf{p}$ can be used to represent the color of the object. Texture can be encoded using gray-value and gradient magnitude. Shape can again be encoded using the distance transform or the fraction of foreground-pixels in a region around $\mathbf{p}$. Similarly, the color and texture features can also be taken as the mean or other statistic in a region around $\mathbf{p}$. Finally, we suggest to normalize each feature to a value in $[0, 1]$. This ensures that the different channels are comparable.

To achieve invariance to rotation and scale, we index pixels relative to a local coordinate system (see Fig. 2.3): Given the center $\mathbf{c}$, major axis $\mathbf{b}_1$ and minor

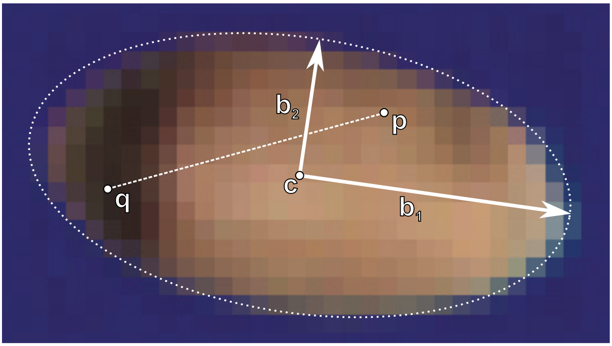

Figure 2.3: Local feature primitives comparing two locations in the image. Each feature is the difference of two descriptors extracted at two locations $\mathbf{p}$ and $\mathbf{q}$ in the object image. Invariance to rotation and scale is achieved by indexing relative to the object center $\mathbf{c}$ and the major and minor axes $\mathbf{b}_1$ and $\mathbf{b}_2$.

axis $\mathbf{b}_2$ of an object, the object indexed coordinates $\mathbf{x} = (\lambda, \mu)$ are transformed to global coordinates as $\mathbf{p} = \mathbf{c} + \lambda\mathbf{b}_1 + \mu\mathbf{b}_2$.

2.2.1 Feature Selection

The definition of these features leaves the question how to to learn discriminative descriptors. For this purpose, we adapt the random fern approach presented by Özuysal et al. [OCLF10]. A random fern F_m can be thought of as a collection of S binary features $f_i \in \{0, 1\}$, which divide the feature space into 2^S disjoint regions $\mathcal{R}_s$. Similar to decision and regression trees each $\mathcal{R}_s$ is associated with an output. In [OCLF10] the output of a single fern is a probability estimate $\hat{p}(c|\mathbf{f})$ that an object belongs to class c given the observations $\mathbf{f}$. The features in each fern are randomly selected. In this report, we propose to instead select the best from a pool of *feature candidates*.

Given a set of N training samples, we randomly sample K local coordinates $\mathbf{x} \in [-1, 1]^2$. Given L different feature extraction methods $\phi^{(l)}$, this produces a pool of $(LK)^2$ feature candidates. The best S features are selected in an iterative scheme, where in each round the feature combination that maximizes the correlation to an output value $\tilde{y}$ (see Algorithm 2.1),

$$\varrho(\phi_i - \phi_k, \tilde{y}) = \frac{\sum_{n=1}^{N}\left(\phi_i^{(n)} - \overline{\phi_i} - \phi_k^{(n)} + \overline{\phi_k}\right)\left(\tilde{y}^{(n)} - \overline{\tilde{y}}\right)}{\sqrt{\left(\sum_{n=1}^{N}\left(\phi_i^{(n)} - \overline{\phi_i} - \phi_k^{(n)} + \overline{\phi_k}\right)^2\right)\left(\sum_{n=1}^{N}\left(\tilde{y}^{(n)} - \overline{\tilde{y}}\right)^2\right)}}.$$

For the sake of brevity we abuse notation and write $\phi_i^{(n)}$ to denote the i-th combination of local coordinate and feature extraction method extracted from the n-th training sample, and $\overline{\phi_i}$ to denote the empirical mean over the samples (likewise for $\tilde{y}$). The threshold τ_n (see eq. (2.1)) is randomly selected in order to be more robust towards non-representative training sets. Finally, the binary feature $f_n = \mathbf{1}[\phi_i - \phi_k < \tau_n]$ is constructed and both ϕ_i and ϕ_k are removed from the pool of candidates. This process is continued until S features are selected.

The resulting random ferns are very fast to compute, but unable to reliably classify unknown objects. However, the combination of multiple ferns was proven to be a reliable classifier [OCLF10].

2.2.2 Fern Boosting

In the original paper, Özuysal et al. combined many random ferns in a semi-naive Bayesian method. However, since we are interested in a minimal number of ferns to keep the computational costs low, we propose to construct a cascade $F_M(\mathbf{x}) = \sum_{m=1}^{M} F_m(\mathbf{x})$ of random ferns using gradient boosting [Fri00]. Following Friedman, we adapt L_2_TreeBoost to employ random ferns instead of regression trees as weak classifiers. The modifications are straightforward, producing Algorithm 2.1.

Given the cascade $F_M(\mathbf{x})$, class membership probability estimates are derived as

$$\hat{p}(y = 1|\mathbf{x}) = \frac{1}{1 + \exp\left(-2F_M(\mathbf{x})\right)} \text{ and}$$
$$\hat{p}(y = -1|\mathbf{x}) = \frac{1}{1 + \exp\left(2F_M(\mathbf{x})\right)}.$$

Objects are classified according to

$$\hat{y}(\mathbf{x}) = 2 \cdot \mathbf{1}\big[c_-\,\hat{p}(y = 1|\mathbf{x}) > c_+\,\hat{p}(y = -1|\mathbf{x})\big] - 1,$$

where c_- and c_+ are the costs of predicting $y = -1$ and $y = 1$ respectively when the true class is $y = 1$ resp. $y = -1$.

3 Conclusion

In this technical report, we proposed two methods of learning discriminative features for automated visual inspection from a sample of wanted and unwanted materials. The first method uses the bag of visual words framework to derive global

Algorithm 2.1 L$_2$_FernBost

Require: Number of iterations M, training set $\{\mathbf{x}^{(n)}, y^{(n)}\}_{n=1}^{N}$ with $y^{(n)} = \pm 1$

$F_0(\mathbf{x}) = \frac{1}{2} \log \frac{1+\overline{y}}{1-\overline{y}}$

for $m = 1, \ldots, M$ **do**

$\tilde{y}^{(n)} = \frac{2y_i}{1+\exp\left(2y^{(n)}F_{m-1}(\mathbf{x}^{(n)})\right)}$ for $n = 1, \ldots, N$

$\{\mathcal{R}_{ms}\}_1^{2^S}$ = random-fern($\{\mathbf{x}^{(n)}, \tilde{y}^{(n)}\}_{n=1}^{N}$) with S features

$\gamma_{ms} = \frac{\sum_{\mathbf{x}^{(n)} \in \mathcal{R}_{ms}} \tilde{y}^{(n)}}{\sum_{\mathbf{x}^{(n)} \in \mathcal{R}_{ms}} |\tilde{y}^{(n)}|(2-|\tilde{y}^{(n)}|)}$ for $s = 1, \ldots, 2^S$

$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \sum_{s=1}^{2^S} \gamma_{ms} \mathbf{1}[\mathbf{x} \in \mathcal{R}_{ms}]$

end for

object descriptors that are invariant to both scale and rotation. The second method combines feature extraction and classification by learning a cascade of random ferns using gradient boosting. The features are sensitive to the defect's location and invariant to object scale and rotation due to the use of a local coordinate system.

Both methods allow to encode all major aspects of object appearance: color texture and shape. Both methods use very primitive underlying features such as the color of a pixel. Since they require only simple operations (sums, products and comparisons), both methods are very fast in operation. This comes at the cost of a lengthy (but automated) training phase.

In the future, we plan to evaluate both methods in different scenarios. The BOV approach is suited when the whole object appearance is of interest, e.g. in grading the ripeness of fruits. The cascade of random ferns will be evaluated on problems where the location of a defect is a major clue, for example in the detection of fungal infections in grains.

Bibliography

[BCGS+09] J. Blasco, S. Cubero, J. Gómez-Sanchís, P. Mira, and E. Moltó. Development of a machine for the automatic sorting of pomegranate (Punica granatum) arils based on computer vision. *Journal of Food Engineering*, 90(1):27–34, January 2009.

[BDLM00] L Bergasa, N Duffy, G Lacey, and M Mazo. Industrial inspection using Gaussian functions in a colour space. *Image and Vision Computing*, 18(12):951–957, September 2000.

[CDF+04] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *International Workshop on Statistical Learning in Computer Vision (ECCV)*, pages 1–22, 2004.

[DCL00] N. Duffy, J. Crowley, and G. Lacey. Object detection using colour. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, volume 1, pages 700–703. IEEE Comput. Soc, 2000.

[DS06] Cheng-Jin Du and Da-Wen Sun. Learning techniques used in computer vision for food quality evaluation: a review. *Journal of Food Engineering*, 72(1):39–55, January 2006.

[Fri00] Jerome H Friedman. Greedy Function Approximation: A Gradient Boosting Machine. *Annals of Statistics*, 29:1189–1232, 2000.

[HPS06] Marko Heikkilä, Matti Pietikäinen, and Cordelia Schmid. Description of interest regions with center-symmetric local binary patterns. In *Computer Vision, Graphics and Image Processing*, pages 58–69. Springer, 2006.

[LCG09] Changyong Li, Qixin Cao, and Feng Guo. A method for color classification of fruits based on machine vision. *WSEAS TRANSACTIONS on SYSTEMS*, 8(2):312–321, February 2009.

[MPZ+03] Elias N Malamas, Euripides G.M Petrakis, Michalis Zervakis, Laurent Petit, and Jean-Didier Legat. A survey on industrial vision systems, applications and tools. *Image and Vision Computing*, 21(2):171–188, February 2003.

[OCLF10] Mustafa Özuysal, Michael Calonder, Vincent Lepetit, and Pascal Fua. Fast keypoint recognition using random ferns. *IEEE transactions on pattern analysis and machine intelligence*, 32(3):448–61, March 2010.

[OPM02] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, July 2002.

[PD07] Florent Perronnin and Christopher Dance. Fisher Kernels on Visual Vocabularies for Image Categorization. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, June 2007.

[RB14] Matthias Richter and Jürgen Beyerer. Parameter-learning for color sorting of bulk materials using genetic algorithms. In *Forum Bildverarbeitung 2014*, pages 107–118. KIT Scientific Publishing, 2014.

[ZLTL14] Dong Zhang, Dah-Jye Lee, Beau J. Tippetts, and Kirt D. Lillywhite. Date maturity and quality evaluation using color distribution analysis and back projection. *Journal of Food Engineering*, 131:161–169, June 2014.