Wooden Knot Detection Using ConvNet Transfer Learning

Rickard Norlander[1], Josef Grahn[2(✉)], and Atsuto Maki[1]

[1] Royal Institute of Technology (KTH), 100 44 Stockholm, Sweden
{norla,atsuto}@kth.se
[2] OptoNova AB, 171 54 Solna, Sweden
josef.grahn@gmail.com

Abstract. This paper presents a method of localizing wooden knots in images of oak boards using deep convolutional networks (ConvNets). In particular, we show that transfer learning from generic images works effectively with a limited amount of available data when training a classifier for this highly specialized problem domain. We compare our method with a previous commercially developed technique based on kernel SVM with local feature descriptors. Our method is found to improve the detection performance significantly: F_1 score 0.750 ± 0.018 vs 0.695. Furthermore, we report some observations regarding the behavior of KL-divergence on the test set which is counter-intuitive in its relation to the accuracy of classification.

Keywords: Detection · Wood · Knots · Convolutional networks · Deep learning · KL-divergence

1 Introduction

Automatic inspection of wooden surfaces is an increasingly important application in the manufacturing process of furniture and flooring[1]. In the forest industry, the most expensive part of running a saw mill is lumber, making up 75% of the cost [24]. The cost of misclassifying material gets greater as we go further in the processing chain since more value is added to the product in each stage. Thus, it is very important not to waste raw material by incorrectly classifying wood; small gains in classifier performance can translate into large cost savings. Detecting knots on wooden surfaces [18,20,22], which we are concerned with in this article, is a critical step in many processes and therefore the detection accuracy is of primary interest. One of the challenges is to cope with the large variation in knot appearances, and it can be difficult to gather sufficient amount of training data with noiseless labeling. Furthermore, the color range of knots completely overlaps with that of normal wood regarding oak material, which makes the task harder.

[1] Around 700,000 cubic meters of wooden boards are produced per year in Sweden alone [23].

R.R. Paulsen and K.S. Pedersen (Eds.): SCIA 2015, LNCS 9127, pp. 263–274, 2015.
DOI: 10.1007/978-3-319-19665-7_22

In the research area of image classification, Krizhesky et al. [14] outperformed the state-of-the-art with a large margin using a new architecture of deep convolutional network (ConvNet) trained with 1.2 millions images [1], and since then ConvNets have become popular and have been successfully applied to different tasks. The ability of deep ConvNets to learn representations has also been studied from various perspectives [4], including transfer learning [16]; the idea there is to learn an efficient generic visual representation by training a ConvNet on a large dataset from one problem domain, then using that network to perform a task in a different problem domain where the amount of labeled data might be smaller. A few authors have recently showed evidence supporting the efficacy of transfer learning in several visual recognition tasks [5,9,21,25].

In this paper, we apply ConvNet transfer learning to the problem of wooden knot detection, where the amount of available training data is limited, with the expectation that the network will yield a performance increase in comparison to the current state-of-the-art. The first contribution of the paper is thereby to show that the new detector based on a ConvNet indeed outperforms the pipeline based on HOG features combined with a kernel SVM classifier despite the limited availability of the training data. Secondly, we will discuss the behavior of Kullback Liebler-divergence (KL-divergence) on the test set when fine-tuning early layers in a network trained with transfer learning.

1.1 Related Work

Transfer learning: Transfer learning is a machine learning approach where data from one problem is used to increase performance in another problem. The approach has previously been used for neural networks and ConvNets, see [2,12,15,19] for a few examples. In [12], the authors trained a ConvNet, viewing it as consisting of two halves, an earlier half and a later half. Transfer of knowledge was achieved by keeping the early layers as-is, and training the later layers for a new task. In the framework of object detection with deep convolutional networks, it has been also shown [6,11] that fine-tuning the network for the target task can help the performance. Our approach is also motivated by the transferability studied in [3].

Knot detection: There have been many efforts described in the literature to find knots in images of wooden surfaces. Some of the previous approaches include [18,20,22]: In an early work on grading wooden board [18] first and second order partial derivatives are computed for each pixel, and used to assign a label such as *edgeeast*, *edgesouth* or *edgenorthwest*. Pixels are then merged into objects. In [22], small neural networks are employed taking gray levels from a 3x3x3 box from a CT-scan, and that box's distance to the center of the tree, as input. Another approach was to search for dark pixels in a photo, merge them into connected components, and remove small ones [20].

The reference method used in this study was developed by OptoNova, and employs HOG descriptor [8] features, which are classified with a soft margin

kernel SVM [7]. To the authors knowledge, the system achieves the quality that is among the highest available in image-based commercial products in this domain.

2 Method

To detect and localize knots in an image of a wooden surface, we use a trained ConvNet to classify overlapping patches in a sliding window fashion at a fixed spatial scale as either *knot* or *wood*. The final layer of the network outputs a confidence value, that we use to signify how likely it is that each respective patch contains a wooden knot. The grid of confidence values can be seen as constituting a "confidence image" over the entire surface. Local maxima reaching above a set threshold in a smoothed version of the confidence image are considered detections.

Algorithm 1. Detection algorithm

procedure DETECT(image J)
  $I \leftarrow$ CROPBACKGROUND(J)
  for all positions (x, y) over I with stride n **do**
    $P_{x,y} \leftarrow$ EXTRACTPATCH(I, x, y)
    $c_{x,y} \leftarrow N(P_{x,y})$ ▷ The network N yields a confidence score
  end for
  $C' =$ GAUSSIANSMOOTH(C, σ) ▷ Where $C = [c_{x,y}]$ is matrix
  $D \leftarrow \varnothing$
  for all $\boldsymbol{p} \in \{(x, y) : c'_{x,y} > t\}$ ordered by descending confidence **do**
    if $\forall \boldsymbol{q} \in D \; ||\boldsymbol{p} - \boldsymbol{q}|| > s$ **then**
      $D \leftarrow D \cup \{\boldsymbol{p}\}$
    end if
  end for
  return D
end procedure

The ConvNet is trained using extracted patches from annotated images. The annotations contain both positive and negative examples, to which additional randomly sampled negatives examples are added. We initialize the network using a network pre-trained on the ImageNet dataset. Before training, the top two layers of this network is replaced by three new layers with random weights.

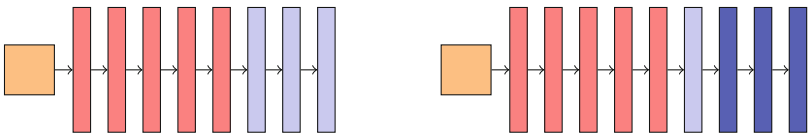

Fig. 1. An ImageNet classifier is turned into a knot classifier by removing the m last layers and replacing them with n new layers. From left to right: image, convolutional layers, fully connected layers, newly added layers

Algorithm 2. Training algorithm

procedure TRAIN(images $\mathcal{J}$, annotations $\mathcal{A}$, pre-trained network M, number-to-remove m, layer-sizes ns)
  for all $J_i \in \mathcal{J}$ **do**
    $I_i \leftarrow$ CROPBACKGROUND(J_i)
    $\mathcal{S}_i \leftarrow \{(\text{EXTRACTPATCH}(I_i, x, y), label) : (J_i, x, y, label) \in \mathcal{A}\}$
    $\mathcal{R}_i \leftarrow \varnothing$
    for n_r random locations (x, y) in I_i **do**
      $P \leftarrow$EXTRACTPATCH(I_i, x, y)
      if $\forall (Q, label) \in \mathcal{S}_i \; P \cap Q = \varnothing$ **then**
        $\mathcal{R}_i \leftarrow \mathcal{R}_i \cup \{(P, \text{'negative'})\}$
      end if
    end for
  end for
  $\mathcal{S} \leftarrow \bigcup(\mathcal{S}_i \cup \mathcal{R}_i)$
  $N_0 \leftarrow$REMOVETOPLAYERS(M, m)
  $N_1 \leftarrow [N_0$ — RELULAYERS(ns) — SOFTMAXLAYER($||\{label\}||$)]
  $N_2 \leftarrow$ SGDTRAINTOPLAYERS(N_1, S)
  $N \leftarrow$ SGDTRAIN(N_2, S)
  return N
end procedure

2.1 Preprocessing: Cropping the Images

We preprocess the images $\{J_i\}$ by doing a background segmentation based on pixel brightness. Using this segmentation, a rectangular crop is applied such that only the wood surface remains. The resulting region is padded with a border of uniform gray, to enable extraction of patches partially outside the image, giving us images I_i.

Fig. 2. Picture of wooden board after cropping

2.2 Training

Creating training examples: From the cropped and padded images $\{I_i\}$, three sets of 300px × 300px patches $P_j \subset I_i$ are extracted: annotated knots $\mathcal{K}$, annotated hard negatives $\mathcal{H}$, and random negative patches $\mathcal{N}$. One class is used for each of these types, meaning that there are two negative classes and one positive class. The random patches are constrained not to overlap with patches from the two other categories.

Training the network: A trained classifier is obtained by taking the *Caffe Reference ImageNet Model*, a pre-trained ConvNet with architecture (showed in Figure 3) very similar to Krizhesky's[14], and fine-tuning it for detecting wooden knots. The transfer learning is done by taking the pre-trained network, removing the last 2 layers, and then adding 3 new layers with random weights at the end (see Figure 1): two ReLU layers with 256 hidden nodes followed by a softmax layer. The mean removal of the ImageNet model is replaced by a subtraction of 128 from all positions and channels[2].

The resulting network architecture has a mean removal layer first, followed by five convolutional layers, in turn followed by three ordinary fully connected hidden layers and then finally a fully connected softmax layer.

The network is trained by stochastic gradient descent with a minibatch size of 64 using the Caffe framework[13]. Random crops of size 227px × 227px are used. During the first 50,000 iterations the network is trained with the original layers frozen, and the remaining using a learning rate of 0.001, and a momentum of 0.9. Dropout is used after every fully connected hidden layer for regularization; the first layer with 50% probability to drop out, and the subsequent two 80%. A weight decay of 0.0005 is used for additional regularization.

The network is then *fine-tuned* by training all layers with a reduced learning rate (a factor 10 lower) and a reduced drop out rate (50% chance for all layers).

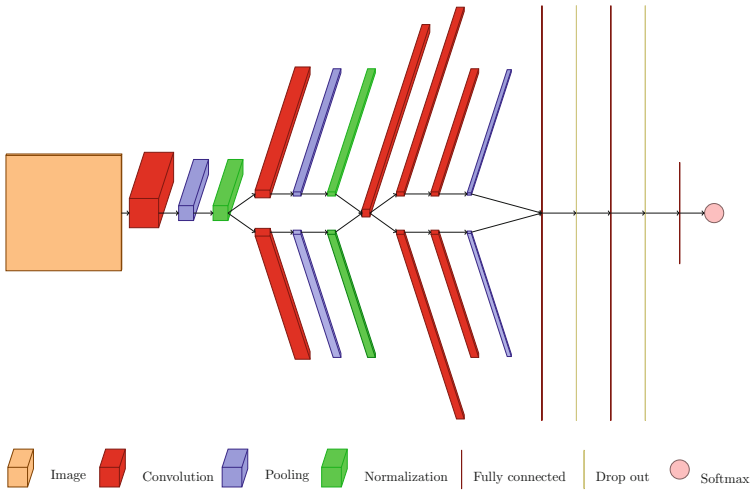

Fig. 3. Architecture of Caffe Reference ImageNet Model. Size of box represents size of data (width × height × channels) after that stage. The fully connected layers have been scaled down by a factor 1/5 to fit.

[2] Subtracting 128 was empirically found to be superior to both subtracting the mean of all patches in the dataset, and subtracting the mean (r,g,b) triplet over the dataset

2.3 Detection

Let I be an image containing an orthographically projected view of a flat wood surface, and let $P_{x,y} \subset I$ denote a 227px $\times$ 227px patch centered at coordinate (x, y) of I. We define the detection confidence $c_{x,y}$ at a coordinate (x, y) in I as $c_{x,y} = N(P_{x,y})$, where $N(P)$ is the output of our classifier N given P as input. This gives us a scalar field $C : (x, y) \mapsto c_{x,y}$, which we call the confidence image of the image I.

A smoothed confidence image is obtained by convolving C with a 2D Gaussian kernel G, yielding $C' = G * C$. Smoothing was done with $\sigma = 26.7$. The confidence image is thresholded to yield a list of detections. Duplicates are removed by going from the strongest to the weakest detection, removing any detection too close to an earlier one.

For run-time efficiency, we apply the classifier with a stride of n pixels in both directions. A stride of $n = 10$ pixels gave a good trade-off between run-time and detection performance according to some quick tests.

3 Experiments

3.1 Dataset

Our dataset consists of 2317 annotated RGB bitmaps of orthographically projected oak boards at a fixed distance, taken with evenly distributed diffuse lighting. Each image is approximately 2500×600 pixels (8 bits per channel), though the size varies somewhat by sample. The relevant oak surface of the boards are surrounded by the visible edges of supporting material and frames, and a dark background, giving us approximately 1800×500 pixels of oak surface per image.

The dataset has 3235 annotated knots and 918 annotated hard negatives. Most of the annotated knots fit within a 200×200 pixel window. The dataset is divided into two parts, a training set (60 %), and a test set (40 %). The test set is used to evaluate the detectors.

3.2 Experimental Design

The classifier was evaluated on the patches of the test set, and also as a detector on the images of the test set. To study the performance of the detector, the detector is applied to all test images. For each detection, an axis parallel square with side 120 px is imagined around it. If there is a knot inside that square, the detection is counted as a true positive, else as a false positive. An undetected knot is a false negative. The F_1 score as a function of threshold is computed, and the maximum F_1 score is used as the score of the detector.

$$score = \max_T F_1(T) = \max_T 2 \cdot \frac{precision(T) \cdot recall(T)}{precision(T) + recall(T)} \tag{1}$$

3.3 Detector Performance

For the final detector, used to compare the ConvNet procedure to the HOG procedure, training was done using an increased amount of random negative patches (24470 patches instead of 2471 as used in other experiments). The best F_1 score for the final ConvNet was 0.750 and for the HOG 0.695. This difference is statistically significant. The precision-recall curves for the two detectors are shown overlaid in Figure 4. The detectors are about tied for lower recalls, with the ConvNet better at mid-high recalls, but being beat at the very highest recalls, and having a worse asymptotic. The precision at low recalls, meaning lower than about 0.2, was very sensitive to parameter choice; sometimes better than HOG, and sometimes worse.

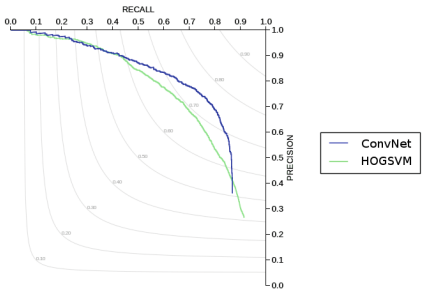

Fig. 4. Precision/recall for detectors shown with F_1-score isocurves

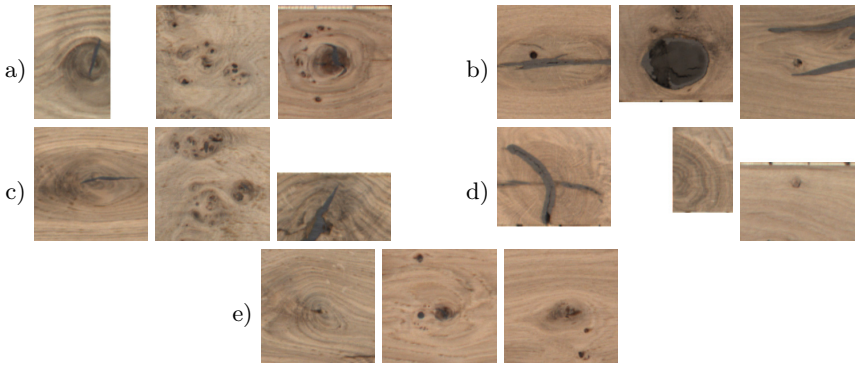

Fig. 5. a) Highest detection strength true positives - ConvNet b) Lowest detection strength false negatives - ConvNet c) Highest detection strength false positives - ConvNet d) Lowest detection strength false negatives - HOG e) Highest detection strength false positives - HOG

3.4 Effects of Transfer Learning

To determine to what degree transfer learning had helped, a classifier was trained without it. The network was trained for 120,000 iterations, after which learning rate and dropout rate was lowered. The network was trained for 40,000 iterations with these new parameters.

Accuracy vs iteration for the model trained from scratch, is displayed in Figure 6. Even after accuracy has converged, it is lower than the accuracy for the pre-trained network.

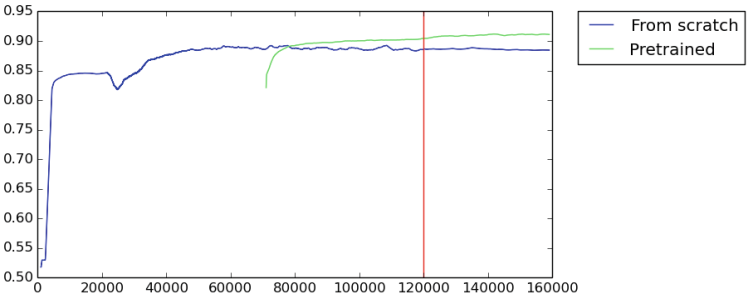

Fig. 6. Accuracy versus iteration for model trained from scratch and pre-trained model. The vertical line signifies start of fine-tuning. Smoothed with symmetric simple moving average of length 2000.

3.5 KL-Divergence During Fine-Tuning

Kullback Liebler-divergence of the training set was used as the optimization objective during training. Interestingly, when fine-tuning the network for our dataset, we observed a drastically worsening KL-divergence score on the test set, while the *classification* error on the test set continued to decrease (see Figure 7).

This means that while the classifier is getting better at *classifying*, it is also getting worse at assigning probabilities. As the probabilities for patches in the training set approach 0 and 1, it seems likely that the same happens on the test set, meaning that those examples it does get wrong, are penalized heavily.

The classifier was contrasted with one fine-tuned differently: learning rate and dropout rate were lowered during fine-tuning like for the normal classifier, but the early layers were not allowed to train. The KL-divergence increases very little. It seems that it is the training of the early layers that causes the issues.

3.6 Significance

In order to obtain a significance estimate of the F_1 score for a given, trained classifier, we employ statistical bootstrapping[10] on the test set. We do this by creating a new test result by sampling with replacement from all false negatives,

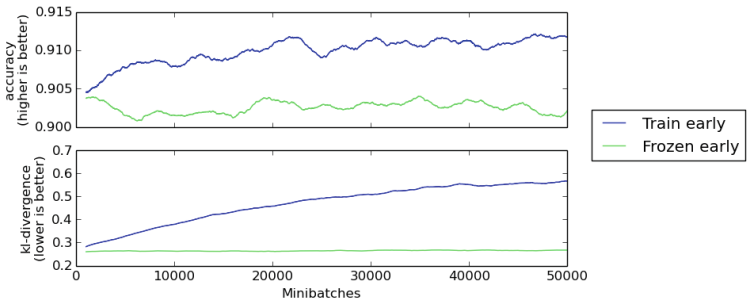

Fig. 7. All curves have been smoothed with symmetric simple moving average of length 2000. Train early: During the fine-tuning, the KL-divergence worsens as the accuracy improves. Frozen early: Lowered learning rate, dropout rate, but frozen early layers. Accuracy is worse, but KL-divergence is better.

false positives and true positives, yielding a new F_1 score, and repeat this process 1000 times. A centered 95 % confidence interval taken from the resulting data series gives a F_1 score uncertainty of ± 0.018.

4 Discussion

Dataset labels: It should be noted that the dataset used in the experiments has a certain degree of inconsistency in the labelings. This can partly be explained by the fuzziness of the underlying classes—knots can be arbitrarily small, and knots transition smoothly to wood when going radially through the trunk of the tree. There are also cases of presumed mislabeling present. This can be seen in the strongest false positive detection in Figure 5. Therefore, the performance measures are not very interesting in themselves but can still be used to compare and contrast across methods.

Transfer learning: Transfer learning from the ImageNet dataset yielded a higher accuracy than when training a network from scratch. In the latter case, training was done for an extended time after the accuracy had reached a stable value (see Figure 6). This indicates that the difference cannot simply be explained by a longer total training time of the pre-trained network. This supports the conclusion that transfer of knowledge does indeed occur, and that it enables features that couldn't otherwise have been learned for lack of data.

In this context, it is worth noting that the ImageNet dataset contains highly diverse data of many different objects in many different poses and configurations, whereas the problem we are transferring to is extremely specialized, and likely should require a much less diverse feature set to represent all possible inputs.

KL-divergence: One thing that was noted during the experiments was that when continuing training with a lower learning rate, the KL-divergence on test data worsened, even as the accuracy on the same data improved (see Figure 7). This was especially pronounced when the early layers were fine-tuned. The poor test KL-divergence means that it can be dangerous to interpret the output of the softmax as a probability. This situation could be corrected by applying probability calibration (see e.g. [17]) to the outputs.

On a more fundamental note, we find it curious that the optimization objective of the training algorithm (KL-divergence), which is meant to act as a proxy for the non-differentiable problem objective (labeling error), diverges on the test set as training progresses. In a sense, deliberately overfitting with regard to KL-divergence produces better test set accuracy for the labeling. This could suggest that there are proxy objectives that are better aligned with the problem of assigning labels than KL-divergence is, which could potentially yield a better classifier if employed during training.

5 Conclusions

We have shown that convolutional networks, and in particular when trained with transfer learning, can be effectively applied to the problem of detecting wooden knots in surfaces of oak wood. Our ConvNet detector, based on a network trained on the ImageNet dataset, outperformed a commercial detector based on conventional feature descriptors and kernel SVM by a statistically significant margin ($F_1 = 0.750 \pm 0.018$ vs 0.695).

By comparing with a network trained from scratch, we showed that transfer learning gives important improvements in this highly specialized problem domain. We also made observations regarding the KL-divergence on the test set during training, in particular that it exhibits a strong overfitting behavior even as the classification accuracy continues to improve.

Acknowledgments. The authors wish to thank Anders Boberg, Ali Sharif Razavian, and the Computer Vision Group of KTH for fruitful discussions. Anders Boberg and Christian Adåker of Optonova provided valuable comments on the paper. The support from the High Performance Computing team at KTH is also appreciated.

References

1. Imagenet large scale visual recognition challenge 2013 (ilsvrc2013). http://www.image-net.org/challenges/LSVRC/2013/
2. Argyriou, A., Evgeniou, T., Pontil, M.: Multi-task feature learning. In: NIPS, pp. 41–48 (2006)
3. Azizpour, H., Razavian, A.S., Sullivan, J., Maki, A., Carlsson, S.: From generic to specific deep representations for visual recognition (2014). arXiv:1406.5774 [cs.CV]
4. Bengio, Y., Courville, A.C., Vincent, P.: Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence **35**(8), 1798–1828 (2013)

5. Bottou, L., Laptev, I., Sivic, J.: Learning and transferring mid-level image representations using convolutional neural networks. In: CVPR (2014)
6. Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A.: Return of the devil in the details: Delving deep into convolutional nets (2014). arxiv:1405.3531 [cs.CV]
7. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning **20**(3), 273–297 (1995)
8. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Schmid, C., Soatto, S., Tomasi, C. (eds.) International Conference on Computer Vision & Pattern Recognition, INRIA Rhône-Alpes, ZIRST-655, av. de l'Europe, Montbonnot-38334, vol. 2, pp. 886–893, June 2005
9. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T.: Decaf: a deep convolutional activation feature for generic visual recognition. In: ICML (2014)
10. Efron, B.: Bootstrap methods: Another look at the jackknife. Ann. Statist. **7**(1), 1–26 (1979)
11. Girshick, R.B., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR (2014)
12. Gutstein, S., Fuentes, O., Freudenthal, E.: Knowledge transfer in deep convolutional neural nets. IJAIT **17**(3), 555–567 (2008)
13. Jia, Y.: Caffe: An open source convolutional architecture for fast feature embedding (2013). http://caffe.berkeleyvision.org/
14. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 25, pp. 1097–1105. Curran Associates Inc. (2012)
15. Li, L.-J., Su, H., Xing, E.P., Li, F.-F.: Object bank: a high-level image representation for scene classification & semantic feature sparsification. In: NIPS (2010)
16. Mesnil, G., Dauphin, Y., Glorot, X., Rifai, S., Bengio, Y., Goodfellow, I.J., Lavoie, E., Muller, X., Desjardins, G., Warde-Farley, D., Vincent, P., Courville, A.C., Bergstra, J.: Unsupervised and transfer learning challenge: a deep learning approach. In: JMLR Proceedings of the ICML Unsupervised and Transfer Learning, vol. 27, pp. 97–110 (2012)
17. Platt, J.C.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: Advances in Large Margin Classifiers, pp. 61–74. MIT Press (1999)
18. Pölzleitner, W., Schwingshakl, G.: Real-time surface grading of profiled wooden boards. Industrial Metrology **2**(3–4), 283–298 (1992). Machine Vision Technology in the Forest Products Industry
19. Pratt, L.Y.: Discriminability-based transfer between neural networks. In: NIPS (1992)
20. Qiu, Z.F.: A Simple Machine Vision System for Improving the Edging and Trimming Operations Performed in Hardwood Sawmills. Master's thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia (1996)
21. Razavian, A.S., Azizpour, H., Sullivan, J., Carlsson, S.: Cnn features off-the-shelf: an astounding baseline for visual recognition. In: CVPR Workshop of DeepVision (2014)
22. Schmoldt, D.L., Li, P., Lynn Abbott, A.: Machine vision using artificial neural networks with local 3d neighborhoods. Computers and Electronics in Agriculture **16**(3), 255–271 (1997)

23. Skogsindustrierna. Skogsindustrin, en faktasamling, 2010 års branschstatistik, 26 11 2014. http://www.skogsindustrierna.org/MediaBinaryLoader.axd?MediaArchive_FileID=62e53e92-510b-4134-a47e-08d6095b2a62&FileName=Faktasamling_Sv_2010.pdf
24. Image Systems. 2013 annual report (2013). http://mb.cision.com/Main/7480/9570148/233985.pdf
25. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. CoRR, abs/1311.2901 (2013)