Applied Artificial Intelligence

An International Journal

ISSN: 0883-9514 (Print) 1087-6545 (Online) Journal homepage: https://www.tandfonline.com/loi/uaai20

Double-stream Convolutional Neural Networks for Machine Vision Inspection of Natural Products

Przemysław Dolata, Mariusz Mrzygłód & Jacek Reiner

To cite this article: Przemysław Dolata, Mariusz Mrzygłód & Jacek Reiner (2017) Double-stream Convolutional Neural Networks for Machine Vision Inspection of Natural Products, Applied Artificial Intelligence, 31:7-8, 643-659, DOI: 10.1080/08839514.2018.1428491

To link to this article: https://doi.org/10.1080/08839514.2018.1428491

Published online: 05 Feb 2018.

APPLIED ARTIFICIAL INTELLIGENCE
2017, VOL. 31, NOS. 7–8, 643–659
https://doi.org/10.1080/08839514.2018.1428491

Taylor & Francis
Taylor & Francis Group

Double-stream Convolutional Neural Networks for Machine Vision Inspection of Natural Products

Przemysław Dolata, Mariusz Mrzygłód, and Jacek Reiner

Centre for Advanced Manufacturing Technologies, Faculty of Mechanical Engineering, Wroclaw University of Science and Technology, Wroclaw, Poland

ABSTRACT

There are known applications of convolutional neural networks to vision inspection of natural products. For many products it is sufficient to acquire and process a single image, but some might require imaging from two sides. Human experts performing quality inspection of malting barley typically only observe one side of each grain, but in doubtful cases look at both sides, intrinsically combining the information. In this paper, we make two contributions. We present a method for determining whether imaging objects from two sides yields performance benefits over single-sided imaging. Then we introduce a double-stream convolutional network for reasoning from two images simultaneously and analyze several methods of combining information from two streams. We find that when orientation of the object is unpredictable and the streams are not specialized to process a particular view, a fully shared architecture combining information on the prediction level yields best performance (98.7% accuracy on our dataset).

ARTICLE HISTORY
Received 15 September 2017
Accepted 12 January 2018

KEYWORDS
Convolutional neural networks; machine vision; natural products; information fusion

Introduction

Modern food processing industry has to guarantee high quality and safety of products, therefore quality inspection is one of the most important and universal operations. Machine vision methods are widely known as successful tools for automating this inspection in manufacturing. But in the food industry source materials are unique in the fact that they are inherently irregular and variable. They cannot be analyzed with the same algorithms as most synthetic goods. Acceptable quality objects have a high intra-class variance of features, while the inter-class variance between good and defective objects is relatively low, making the task of automating the inspection challenging.

An example of such product is barley grain, a key ingredient in production of beer and whiskey, with high quality requirements. In current industrial practice, identifying defects and contaminations of barley still has to be done visually, based on statistical sampling. A human expert usually looks at multiple grains scattered over a flat surface and hand-picks the defective ones. In case when a

CONTACT Przemysław Dolata ✉ przemyslaw.dolata@pwr.edu.pl Faculty of Mechanical Engineering, Wyb. Wyspianskiego 27, 50-370 Wroclaw, Poland.

Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/UAAI.

decision cannot be made with enough confidence, an individual grain is picked up and viewed with more detail, including rotating to reveal its opposite side. This means that a human is able to perform reasoning based on combined information from multiple views.

Existing approaches to machine vision inspection of samples of barley grain rely on typical document scanners for data acquisition and digital image processing techniques such as morphological operations, edge detection, or segmentation. In these works, arbitrary sets of shape, color, and texture features are extracted from images, so that each grain is represented with a feature vector. Those vectors are then used to train classifiers such as Linear Discriminant Analysis (Zapotoczny, Zielinska, and Nita 2008), artificial neural networks (Nowakowski et al. 2012), hyperellipsoidal decision boundary classifiers (Szczypiński, Klepaczko, and Zapotoczny 2015), or k nearest neighbors (Hailu and Meshesha 2016). However, none of them consider simultaneous processing and reasoning from more than one image of a grain, despite (Szczypiński and Zapotoczny 2012) suggesting that for varietal recognition, both dorsal and ventral sides should be analyzed.

Over the last few years, a new branch of machine learning known as deep learning has improved the state-of-the-art in computer vision. Convolutional neural networks (CNNs) have been particularly successful at tasks such as image classification and object detection. In classical machine learning, a number of arbitrary, hand-engineered features is extracted from image, and then supplied to a trainable classifier. Performance of the model is thus limited by the quality of detected features. The principle behind deep learning is that both classification function and the feature extraction transformation are trained. There is no arbitrariness of a designer (human) and the model learns the most relevant features directly from data.

One of the earliest industrial applications of deep learning was recognition of handwritten postal codes (LeCun et al. 1989). Since then several more complex industrial problems have been solved using CNNs of various architectures—Deng (2014) presents a detailed applications survey. Recent years brought deep learning solutions to natural objects inspection problems (Brahimi, Boukhalfa, and Moussaoui 2017; Grinblat et al. 2016; Lee et al. 2015; Potena, Nardi, and Pretto 2016; Reyes, Caicedo, and Camargo 2015; Sladojevic et al. 2016; Sünderhauf et al. 2014).

The concept of reasoning from multiple images or using multiple classifiers has been researched in the context of deep learning, starting with Cireşan, Meier, and Schmidhuber (2012) who used ensemble methods for road sign recognition. More recently, Scott et al. (2017) proposed a prediction-level fusion of multiple CNN classifiers processing a single image. Other works (Lin, RoyChowdhury, and Maji 2015; Liu et al. 2017), particularly concerning facial recognition (Lu et al. 2017; Xiong et al. 2016), extract different kinds of features and perform classification basing on a fused representation. This approach has been found successful in processing images of different modalities (Audebert, Saux, and Lefèvre 2017; Su et al. 2015), particularly in video analysis and action recognition (Park et al. 2016). In this area however, prediction-level fusion is also being researched (Li, Chen, and Hu 2017; Simonyan and Zisserman 2014; Ye et al. 2015). Snoek et al. (2005) provide a comparison of early versus late fusion approaches, but their results are not definitely conclusive.

This study is a part of a project to design a complete, automatic barley grain inspection system, capable of imaging individual grains and separating them basing on defects. Whether to equip the system with a single camera or a pair of cameras is an important design choice, influencing costs of the finished device. If imaging the grain from both sides improves classification accuracy, this additional expense will be justified. In this case, the most effective method of combining information from both images needs to be developed.

In this paper, we apply a CNN to barley defect recognition problem, introducing a novel double-stream architecture for simultaneous processing of both sides of a grain. We present our contribution in two distinct sections. In the first, we provide a problem-independent method of examining whether double-sided imaging really increases classification performance of the system. In the second, we analyze several methods of combining information from the dual streams, comparing architectures with different fusion points.

Convolutional neural networks

In computer vision input data are high-dimensional, every pixel of an image constituting an independent dimension. For this reason, classical machine learning algorithms do not input pixel data directly. Instead, images are first processed with feature extraction algorithms in order to reduce representation to a less dimensional form. The difficulty lies in designing this extraction algorithm, so that it provided meaningful information about samples, allowing distinguishing classes in the problem space. CNNs alleviate this difficulty by learning to perform feature extraction.

Connections in those networks are arranged in special patterns, instead of full connectivity between units of subsequent layers. Each unit is only connected to some region of the input space (receptive field). Weights are also constrained, so that units with different receptive fields share the same weights. Effectively, a single layer of a CNN performs a convolution operation on its input, with convolution kernel being a trainable parameter. Stacking convolutional layers with activation functions allows construction of deep networks which learn to detect a hierarchy of features—from simple ones, like edges or color gradients, to more abstract patterns. Those deep CNNs are usually followed by fully connected neural networks (FC) in order to perform classification or regression. This way both the classification and feature detection functions can be jointly learned directly from data, using known gradient descent algorithms.

There are several unique problems regarding CNNs. The convolution alone does not reduce dimensionality effectively enough—for this reason *pooling* is typically used. Making the networks deep by stacking many layers of neurons is the crucial factor in their performance, but gradients tend to vanish at saturating activation functions. As a solution, a new type of nonlinearity, the rectified linear unit (ReLU), has been introduced (Krizhevsky, Sutskever, and Hinton 2012). Several methods of normalizing responses between multiple convolution kernels have been developed (Ioffe and Szegedy 2015; Krizhevsky, Sutskever, and Hinton 2012). Dropout has been proposed as a countermeasure to overfitting and co-adaptation of units in large networks (Srivastava et al. 2014). Normalization of network outputs into a probability distribution is typically done with the softmax function.

Training a network, assuming each sample belongs to only one of the possible classes, can be understood as minimizing a multinomial logistic loss function:

$$E = -\frac{1}{N}\sum_{i=1}^{N}\log\left(p_{i,l}\right)$$

where N is the number of examples in a batch and $p_{i,l}$ is the probability assigned to the correct label for i-th example.

The standard gradient descent is not used to train CNNs: calculating the exact gradient of the loss function is computationally prohibitive due to large number of parameters to optimize and typically large amount of training data. Instead, gradient from only a small portion of the training data (batch) is computed and used to update weights—this is referred to as stochastic gradient descent (SGD).

Experimental setup

Training deep CNNs is still a computationally heavy task and in order to do it efficiently an optimized software implementation is necessary. We chose to use the Caffe framework (Jia et al. 2014), due to its high performance GPU implementation via cuDNN and a freely available repository of pretrained models, the ModelZoo.

Primary CNN architecture used throughout this study is based on AlexNet (Krizhevsky, Sutskever, and Hinton 2012), consisting of five convolutional and three fully connected layers. It was originally designed for the ImageNet classification task (Jia Deng et al. 2009), which is by two orders of magnitude larger in terms of amount of data and number of classes than our problem. Thus we decided to reduce the number of neurons in fully-connected layers from 4096 to

512 neurons to reduce potential of overfitting in the classifier. Additionally, we initialize weights in convolutional layers by transferring them from a model pretrained on ImageNet in order to increase detection performance and limit overfitting in that part of the network.

Double-stream networks

In the case of double-sided imaging, each sample would be given by a pair of images. A naïve solution would be to process each image separately and combine those predictions externally. Xu, Krzyzak, and Suen (1992) give several methods of combining classifiers. But a human expert can view a single grain from many angles during the reasoning process itself, instead of rating two separate images. Intuitively, a network simultaneously processing two images might learn correlations between features found in each of them, potentially providing more accurate predictions at test time.

Since both images are of the same modality (RGB data), they should be processed using the same architecture. Therefore, we decide to duplicate layers from the baseline model in order to create a double-stream network. This network will take a pair of images as input, feeding each one to a corresponding stream. Computation will take place independently for both streams, until the merging point where the information will be combined. The actual choice of this merging point and method of fusion is dependent on the architecture and will be discussed in respective section.

We make no assumptions regarding the arrangement of images—that is, there are no guarantees that, for example, a dorsal side view will always be in the first image of a pair. The network must be robust to randomness in arrangement. We accomplish that by sharing weights and gradients between the duplicated layers, enforcing that their parameters will always be equal. This has the following technical consequence. One training iteration in a single-stream network results in one update of its weights (the following assumes training with simple SGD with momentum):

$$V_t \leftarrow \mu V_{t-1} - \alpha \nabla L(W_{t-1})$$

$$W_t \leftarrow W_{t-1} + V_t$$

where W_i is the weights vector in iteration i, V_i is the weight updates vector in iteration i, $\nabla L(W)$ is gradient w.r.t. the weights W and μ, α are learning hyperparameters: momentum and learning rate. In a double-stream network, each stream has its own, different gradient $\nabla L^{(1)}$ and $\nabla L^{(2)}$, but the update is still applied to a single weights vector:

$$V_t \leftarrow \mu V_{t-1} - \alpha \nabla L^{(1)}(W_{t-1}) - \alpha \nabla L^{(2)}(W_{t-1})$$

$$W_t \leftarrow W_{t-1} + V_t$$

This is effectively the same as doubling the batch size for this network, making the training more aggressive. In case of a network that consists of a shared double-stream part merged into a single stream, this makes the shared part learn faster than the single-stream part.

Output of every network will be normalized into a probability distribution using the softmax function. Those distributions will be transformed into correct or false answers differently in two settings: binary classification (grain defective or acceptable) and multi-class classification (index of exact defect or acceptable class). For the multi-class task, probabilities will be converted into indices by choosing the highest ranked element (argmax). For binary classification, the probabilities corresponding to defect classes will be summed and a sample will be marked as acceptable if this collective defect probability will be lower than some threshold. This gives a degree of control over the classifier, allowing tuning of its sensitivity and precision, but is infeasible in cases where no preferred class can be specified (e.g. varietal classification). Additionally, we analyze the binary classifier performance in the case of simply choosing the most probable element (argmax).

Data acquisition and preprocessing

Malting barley grains were obtained from malting house supply and segregated by an expert into seven classes, corresponding to healthy grains and six types of defects and impurities as specified by the Polish Standard (Polish Committee for Standardization 1998).

In order to acquire images of many grains efficiently, a prototype automatic imaging system is used (Lampa, Mrzygłód, and Reiner 2016). Barley is imaged by two color CCD cameras with lenses and strobed LED ring illuminators, installed coaxially above and below a transparent plate (Figure 1). Grains transportation through system is ensured by vibratory and screw feeders and rotation of the plate. Due to additional barriers fixed above the plate each grain is centered in cameras' fields of view. This allows for acquisition of two images per grain: one showing its ventral, another its dorsal side, with efficiency of up to 8 grains per second. However, at such processing speed it is not possible to force the grains to assume any desired orientation in the imaging area—consequently, there is no information which side of the grain is shown in which image.

Images acquired in this process are not modified in any way except for a crop to a square of 800 × 800 pixels, centered on a grain's visual center of mass, and resize to 227 × 227 pixels with bilinear interpolation. Directly before putting into the neural network, a mean image of the entire dataset is subtracted from each image, in order to zero-center the data.

We acquired 18,463 pairs of images in seven classes ranging from 160 to 7753 pairs in each class. Figure 2 shows examples of several defects and a good quality grain (bottom most row) for comparison. Data were randomly

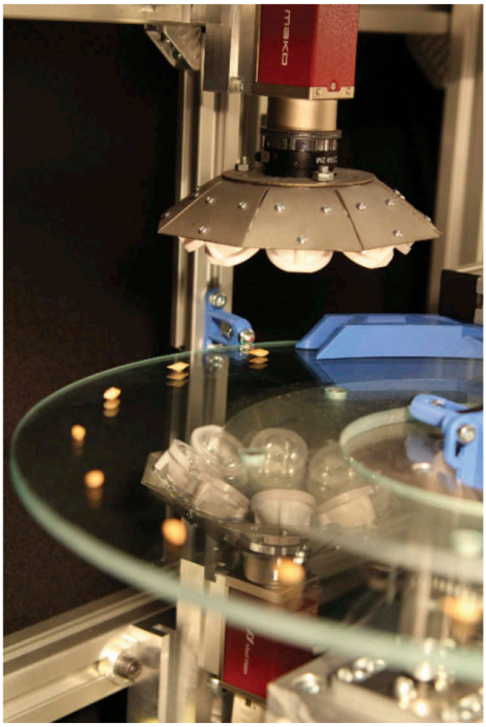

Figure 1. Double-sided image acquisition system.

partitioned into three complementary, approximately equal bins, preserving the class size proportions. These bins were then assembled into three cross-validation folds: each time a different bin was selected as training dataset, with the two remaining ones constituting a validation set.

Comparison of single-stream and double-stream architectures

In this section we demonstrate a method of comparing, whether imaging objects from two sides provides any classification performance benefit over imaging only one side. In the case of double-sided imaging, the resulting dataset would be twice as large as in the latter case, giving the double-sided approach the advantage of more training data. We attempt to isolate the improvement introduced by imaging method itself from the influence of increased dataset size.

For the default double-stream architecture we choose to duplicate the convolutional part of the network, join the extracted features using simple concatenation, and train a fully-connected classifier on that representation. This however introduces the earlier described problem of gradient flow inequality through shared and non-shared parts of the network. In order to provide equal conditions for comparison of single-stream and double-stream architectures, we apply the following procedure (Figure 3):

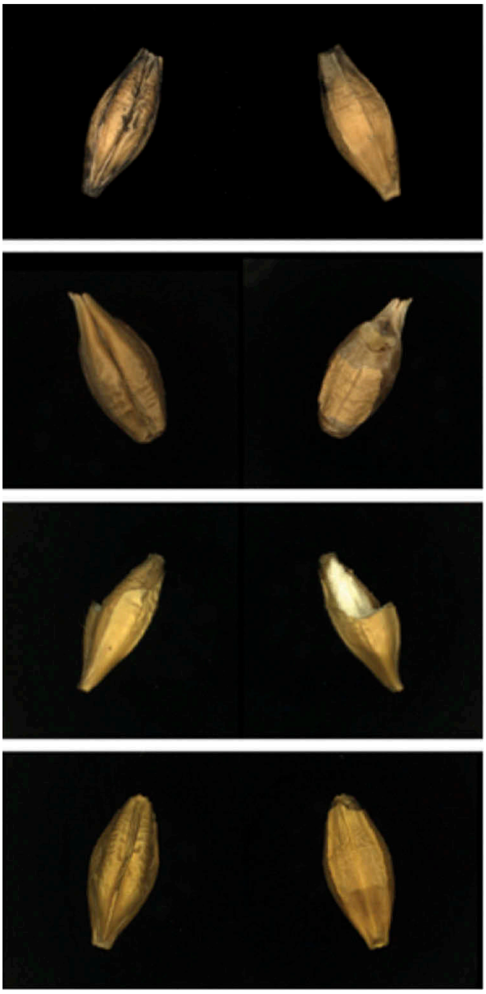

Figure 2. Examples of pairs of acquired images.

- First we train a dummy single-stream network on a dataset consisting of an unordered sequence of images from both cameras (no information about dorsal/ventral correlation is provided at this stage).
- Then we extract weights of its convolutional part and use them to initialize the single-stream and both of the double-stream convolutional parts. Subsequently, we freeze those layers, that is: prevent them from further updates—thus making sure both networks will extract the same features. This will also alleviate the uneven gradient flow in the double-stream architecture.
- Finally we proceed to train the networks, exposing the double-stream model to all pairs of the training set, but picking only one image from each pair for the single-stream network.

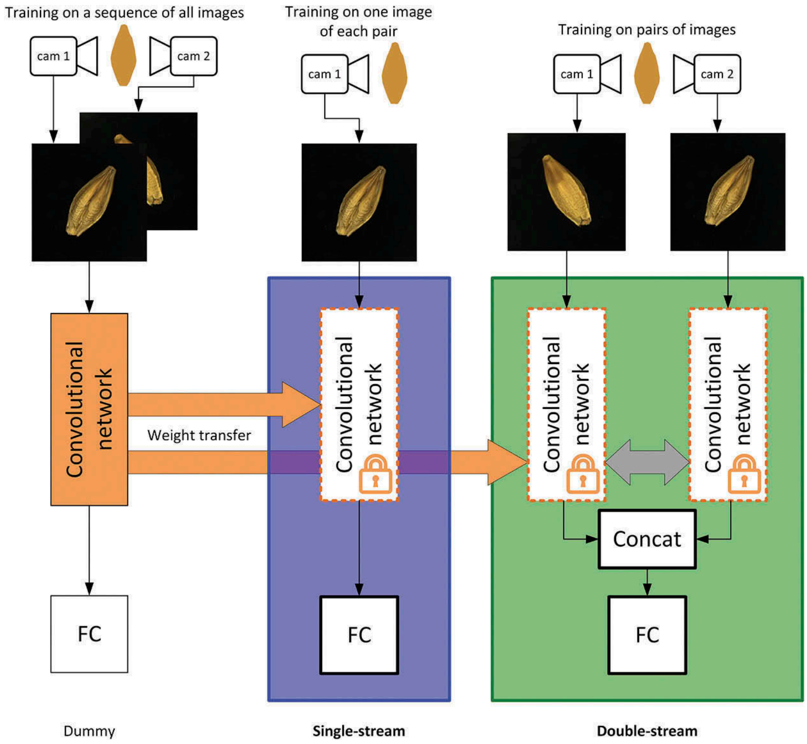

Figure 3. Training procedure.

This ensures equal exposition of feature extractors to training samples, while allowing the classifiers to only learn from the amount of data they would be given during normal operation.

Training hyper-parameters for both networks are the same: we use the standard SGD algorithm with momentum term $\mu = 0.9$, batch size 128 and initial learning rate $\alpha = 0.005$, which is divided by 10 after the first 10 training epochs, and again after the next 3. Results after the total of 15 epochs, averaged between three runs for each model on each of the cross-validation folds, are shown in Table 1.

Figure 4 shows obtained classification accuracy (continuous lines) achieved on the validation set by both models as a function of the threshold value. The maximum accuracy point is marked on each curve. They are additionally compared with performance of the argmax method, shown as a dashed line in the same color as the corresponding curve.

Figure 5 shows the ROC curves for both models, that is plots of false positive ratio on x axis and true positive ratio on the y axis as functions of threshold value, centered on the most interesting region. Performance at the threshold value corresponding to the best binary accuracy is marked with a full circle on each curve. Performance of argmax classifiers is marked with a star. Each point is drawn in the same color as the corresponding curve.

Table 1. Results of the single-stream vs double-stream comparison.

Task	Binary classification				Multi-class clasification
Decision method	Argmax			Threshold (best)	Argmax
Measure	Accuracy	TPR	FPR	Accuracy	Accuracy
Single-stream	97.95%	98.07%	2.21%	98.03%	96.50%
Double-stream	98.18%	98.17%	1.81%	98.22%	96.82%

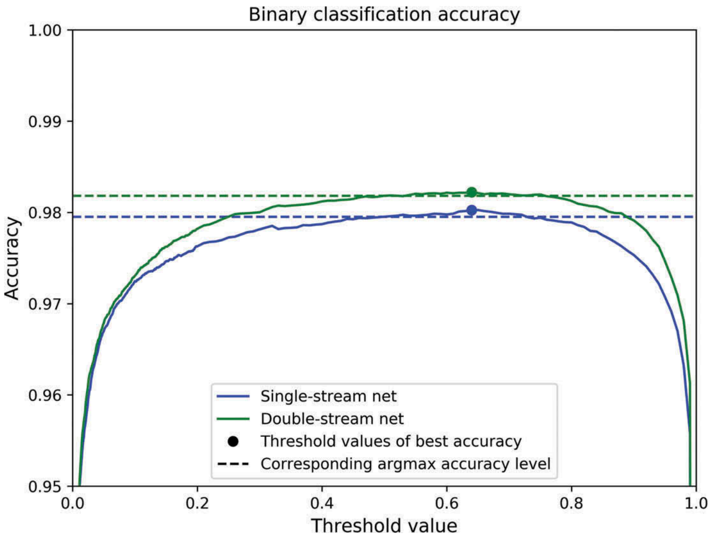

Figure 4. Binary classification accuracy plotted as function of threshold value.

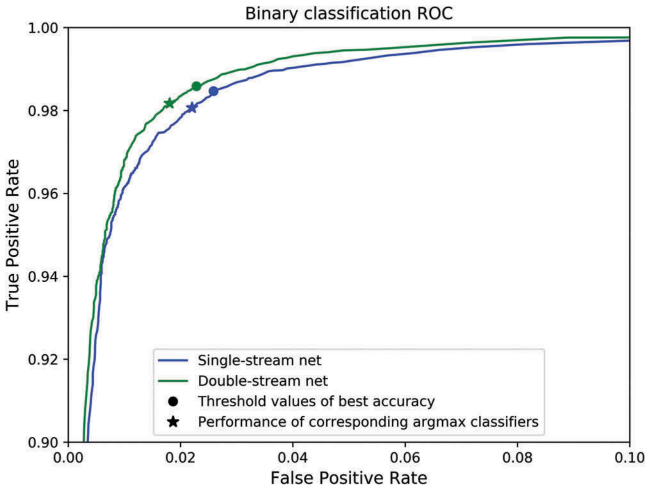

Figure 5. Binary classification ROC with argmax performance for reference.

Double-stream architecture outperforms the single-stream one by 0.2% in binary classification task, regardless of method of decision-making, and by over 0.3% in multi-class task. Analysis of the ROC proves that the double-stream yielded better results at every setting of the threshold. The double-stream model made less errors of the first type (false alarms).

It is worth repeating, that results of this experiment will be task-specific, depending on the data. For the case of machine vision inspection of barley grain, we however conclude that imaging objects from both sides will indeed improve classification performance.

Combining information in double-stream convolutional neural networks

Here we present and compare several methods for joining information from two convolutional network streams. We begin with the early fusion model (Figure 6a) and the view-pooling method of Su et al. (2015) (Figure 6b). In both of them the convolutional parts are shared and their outputs are joined before the fully connected classifier. In the early fusion model this is realized by simple concatenation and in the view-pooling method by an element-wise max operation (Eq. 1).

$$y_i = \max(a_i, b_i) \tag{1}$$

In the next architecture, the late fusion (Figure 6c), streams are not joined until the final classifier layer. The convolutional parts and first two of the fully-connected layers are shared, their outputs are concatenated and this intermediate vector is an input to the final layer.

In this experiment, it is assumed that all models will be trained and deployed on pairs of images, so there is no need for special conditions ensuring equal comparison. Therefore, those three models can be learned jointly in a uniform setting, the same as in the previous experiment.

This is not the case with the prediction-level fusion. In this architecture the entire networks are shared, including the classifiers which are however stripped of the softmax function. Raw fully-connected outputs are concatenated and input to an additional classifying layer. This model is trained in two stages: first the shared streams are trained as single networks without any fusion (Figure 6d), then they are frozen and the additional classifier is trained on their outputs (Figure 6e).

These architectures are finally compared to the naïve approach. This is to train only one network like in the first stage of the prediction-level fusion model (Figure 6d), and test it on two images separately, performing the fusion of predictions for each image externally (Figure 6f). This operation can be performed in several ways, depending on the decision mode (e.g. request that both streams assign at least some minimum confidence to the preferred class, average the distributions). We have found that simply calculating the average of two outputs yields the best performance, therefore we chose to use this method in comparisons.

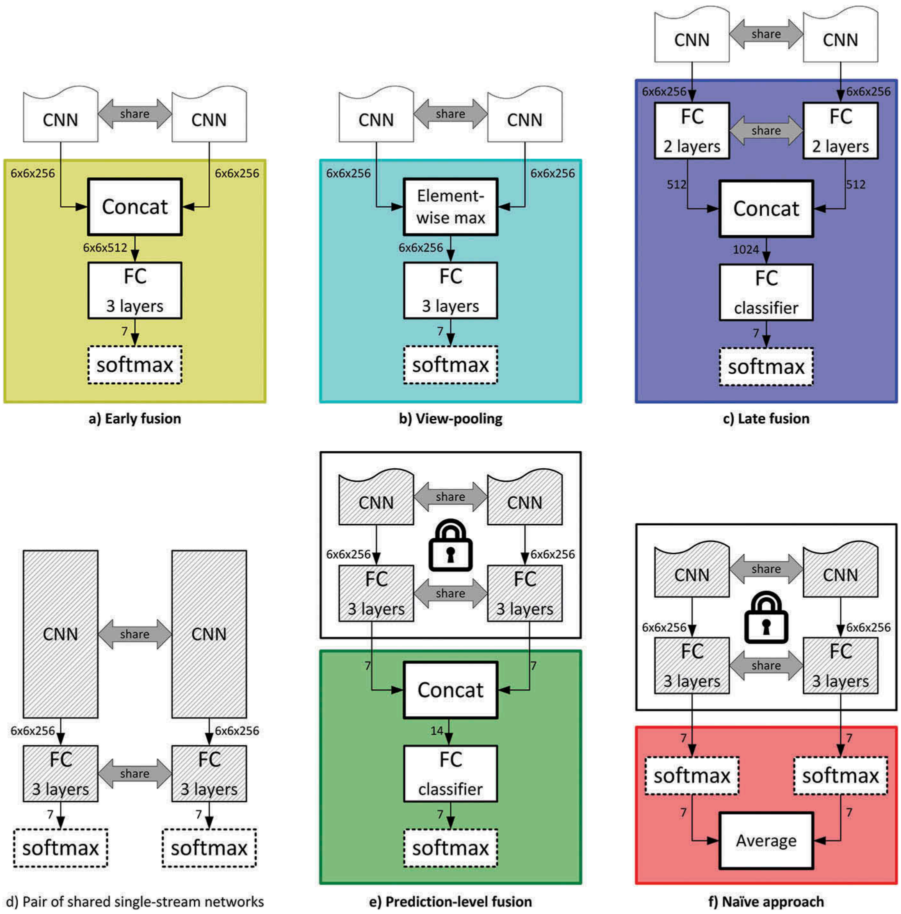

Figure 6. Fusion architectures.

Table 2 summarizes performance of the trained models, as tested on respective validation sets and averaged over three cross-validation folds. Figure 7 shows a plot of binary accuracy versus threshold setting along with argmax performance while Figure 8 compares ROC. Both figures are organized as in the previous experiment.

Architectures with late fusion perform generally better than those with early fusion. Models with convolutional feature fusion are significantly less accurate than those where merging occurs near or after the classifier, as much as 0.6% less accuracy in binary classification and 1.5% less in the multi-class task. The view-pooling method yielded worst results. We suppose this is due to the element-wise max operation only propagating gradients to the stream which produced the maximum output. This means that gradients are split between streams instead of shared, effectively slowing down the learning for that part of the model.

Surprisingly, the naïve classifier performs on par with the best, prediction-level model, equaling its performance in binary task and surpassing it by 0.1% in

Table 2. Results of comparison of various fusion models, best results in bold.

Task	Binary classification				Multi-class clasification
Decision method	Argmax			Threshold	Argmax
Measure	Accuracy	TPR	FPR	Accuracy	Accuracy
Early fusion	98.23%	98.27%	1.83%	98.26%	96.26%
View pooling	98.10%	98.10%	1.90%	98.15%	96.08%
Late fusion	98.58%	98.79%	1.70%	98.60%	97.28%
Prediction-level f.	98.74%	98.88%	1.46%	98.74%	97.61%
Naïve approach	98.72%	98.85%	1.46%	98.73%	97.70%

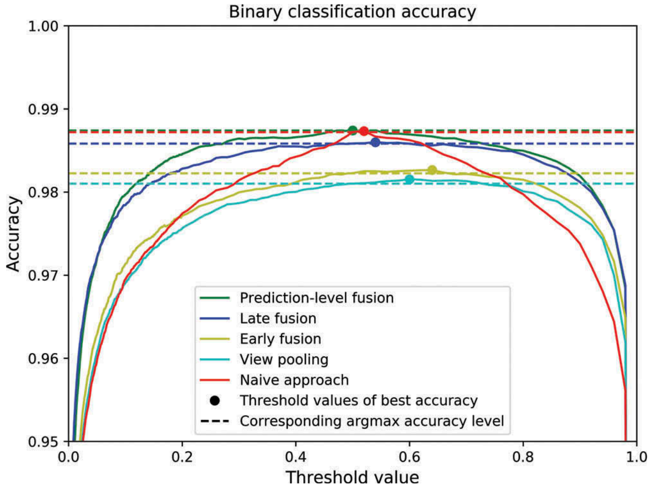

Figure 7. Binary classification accuracy plotted as function of threshold value.

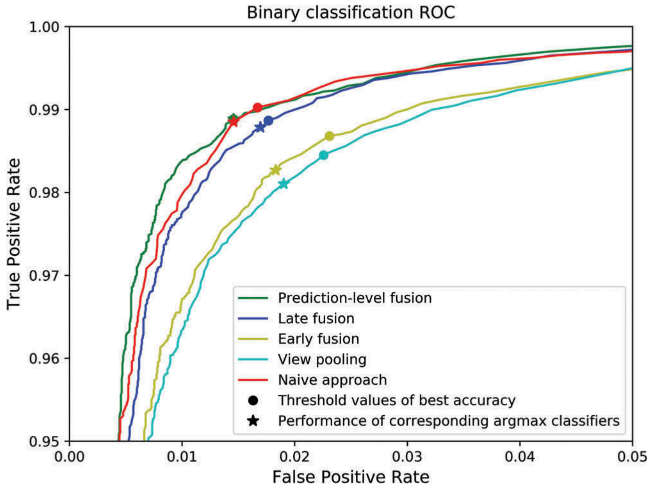

Figure 8. Binary classification ROC with argmax performance for reference.

multi-class. This might mean that none of the double-stream architectures were able to learn any conditional relationships between features extracted from two images. It could be due to the fact that images are fed into the networks with no predictable arrangement—results would have been different if either the grains could be forced to assume a specific orientation in the imaging device, or this orientation could be recognized in a preprocessing step. However, in an industrial environment requiring high processing speed, this might not be feasible.

Accuracy plots in binary classification with threshold decision mode reveal the primary weakness of the naïve approach. While the double-stream network with prediction-level fusion remains highly accurate over a much wider range of values, the naïve classifier only has a single accuracy peak—its performance quickly degrades in both directions. This means that the double-stream architecture gives more freedom in setting the cut-off point and allows more customization of balance between sensitivity and precision of the classifier.

Conclusions

We introduced a double-stream CNN architecture for double-sided machine vision inspection of natural products on the example of barley grains. We presented a method of verifying, whether such approach yields classification performance benefit, separating the improved architecture factor from the influence of simply increased amount of training data. For the task of barley classification, it proved that double-sided imaging results in better performance (98.22% vs 98.03% for single-sided approach).

We analyzed several information fusion models, testing them on the task of barley grain classification. We found that the best double-stream architecture with prediction-level fusion performs almost equally well as a naïve approach of averaging two independent predictions. Both models achieved 98.74% accuracy in binary and, respectively, 97.61% and 97.70% in multi-class classification. Still, the double-stream network, due to a more stable accuracy characteristic, allows a greater freedom of tuning the binary classifier.

The reason that the most straightforward approach was as effective as the complex double-stream network might have been the fact that images were assigned to the streams in a random order. Each side of the grain looks differently, potentially containing different features to extract. An alternative approach would be to use a model consisting of two specialized networks to process dorsal and ventral sides separately, together with a preprocessing step to identify the orientation of a grain. This might provide performance improvement over a model processing both images agnostic of their arrangement.

Acknowledgements

We wish to thank Piotr Lampa for performing data acquisition. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors. However, the source material (barley grain) for the dataset preparation was supplied from a project financed by National Centre for Research and Development—Project PBS3/A8/38/2015.

Funding

National Centre for Research and Development—Project PBS3/A8/38/2015

References

Audebert, N., B. Le Saux, and S. Lefèvre, 2017. Fusion of heterogeneous data in convolutional networks for urban semantic labeling. Presented at 2017 Joint Urban Remote Sensing Event (JURSE), Dubai, UAE, pp. 1–4. doi: 10.1109/JURSE.2017.7924566

Brahimi, M., K. Boukhalfa, and A. Moussaoui. 2017. Fusion of heterogeneous data in convolutional networks for urban semantic labeling. Presented at 2017 Joint Urban Remote Sensing Event (JURSE), Dubai, UAE, pp. 1–4. doi:10.1109/JURSE.2017.7924566.

Cireşan, D., U. Meier, and J. Schmidhuber. 2012. Multi-column deep neural networks for image classification. Presented at 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, pp. 3642–3649. doi: 10.1109/CVPR.2012.6248110.

Deng J., W. Dong, R. Socher, L. J. Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. Presented at 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, pp. 248–255. doi: 10.1109/CVPR.2009.5206848

Deng, L. 2014. A tutorial survey of architectures, algorithms, and applications for deep learning. *APSIPA Transactions on Signal and Information Processing*, 3. doi:10.1017/atsip.2013.9

Grinblat, G. L., L. C. Uzal, M. G. Larese, and P. M. Granitto. 2016. Deep learning for plant identification using vein morphological patterns. *Computers and Electronics in Agriculture* 127:418–24. doi:10.1016/j.compag.2016.07.003.

Hailu, B., and M. Meshesha, 2016. Applying image processing for malt-barley seed identification. Presented at the Conference: Ethiopian the 9th ICT Annual Conference 2016 (EICTAC 2016), Addis Ababa.

Ioffe, S., and C. Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *Proceedings of the 32nd International Conference on Machine Learning*, 37:448–456

Jia, Y., E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. 2014. Caffe: Convolutional Architecture for Fast Feature Embedding. Proceedings of the 22nd ACM international conference on Multimedia (MM '14). ACM, New York, NY, USA, 675–678. doi: 10.1145/2647868.2654889

Krizhevsky, A., I. Sutskever, and G. E. Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems* 25:1097–1105

Lampa, P., M. Mrzygłód, and J. Reiner. 2016. Methods of manipulation and image acquisition of natural products on the example of cereal grains. Control & Cybernetics 45 (3):339–354

LeCun, Y., B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. 1989. Backpropagation applied to handwritten zip code recognition. *Neural Computation* 1 (4):541–51. doi:10.1162/neco.1989.1.4.541.

Lee, S. H., C. S. Chan, P. Wilkin, and P. Remagnino, 2015. Deep-plant: Plant identification with convolutional neural networks, in: 2015 IEEE international conference on image processing (ICIP). Presented at the 2015 IEEE International Conference on Image Processing (ICIP), pp. 452–56. doi:10.1109/ICIP.2015.7350839

Li, H., J. Chen, and R. Hu. 2017. Multiple feature fusion in convolutional neural networks for action recognition. *Wuhan Univ. J. Nat. Sci.* 22:73–78. doi:10.1007/s11859-017-1219-4.

Lin, T.-Y., A. RoyChowdhury, and S. Maji, 2015. Bilinear CNNs for fine-grained visual recognition, in: Proceedings of the IEEE International Conference on Computer Vision. pp. 1449–57.

Liu, Y., X. Chen, H. Peng, and Z. Wang. 2017. Multi-focus image fusion with a deep convolutional neural network. *Information Fusion* 36:191–207. doi:10.1016/j.inffus.2016.12.001.

Lu, X., X. Duan, X. Mao, Y. Li, and X. Zhang. 2017. Feature extraction and fusion using deep convolutional neural networks for face detection. *Mathematical Problems in Engineering* 2017:1–9. doi:10.1155/2017/1376726.

Nowakowski, K., P. Boniecki, R. J. Tomczak, S. Kujawa, and B. Raba, 2012. Identification of malting barley varieties using computer image analysis and artificial neural networks. Presented at the Fourth International Conference on Digital Image Processing (ICDIP 2012), International Society for Optics and Photonics, p. 833425. doi:10.1117/12.954155

Park, E., X. Han, T. L. Berg, and A. C. Berg, 2016. Combining multiple sources of knowledge in deep CNNs for action recognition, in: 2016 IEEE Winter Conference on Applications of Computer Vision (WACV). Presented at the 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1–8. doi:10.1109/WACV.2016.7477589

Polish Committee for Standardization, 1998. PN-R-74110:1998P. Barley – Test methods.

Potena, C., D. Nardi, and A. Pretto, 2016. Fast and accurate crop and weed identification with summarized train sets for precision agriculture, in: Intelligent Autonomous Systems 14. Presented at the International Conference on Intelligent Autonomous Systems, Springer, Cham, pp. 105–21. doi:10.1007/978-3-319-48036-7_9

Reyes, A. K., J. C. Caicedo, and J. E. Camargo, 2015. Fine-tuning deep convolutional networks for plant recognition, in: Working Notes of CLEF 2015 Conference.

Scott, G. J., R. A. Marcum, C. H. Davis and T. W. Nivin. 2017. Fusion of Deep Convolutional Neural Networks for Land Cover Classification of High-Resolution Imagery. IEEE Geoscience and Remote Sensing Letters 14 (9):1638–1642. doi: 10.1109/LGRS.2017.2722988

Simonyan, K., and A. Zisserman. 2014. Two-Stream Convolutional Networks for Action Recognition in Videos. *Advances in Neural Information Processing Systems* 27:568–576

Sladojevic, S., M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic. 2016. Deep neural networks based recognition of plant diseases by leaf image classification. *Computational Intelligence and Neuroscience* 2016:1–11. doi:10.1155/2016/3289801.

Snoek, C. G. M., M. Worring, A. W. M. Smeulders. 2005. Early Versus Late Fusion in Semantic Video Analysis. In: Proceedings of the 13th Annual ACM International Conference on Multimedia, pp.399–402. doi: 10.1145/1101149.1101236

Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15:1929–58.

Su, H., S. Maji, E. Kalogerakis, and E. Learned-Miller, 2015. Multi-view convolutional neural networks for 3d shape recognition. Presented at the Proceedings of the IEEE International Conference on Computer Vision, pp. 945–53.

Sünderhauf, N., C. McCool, B. Upcroft, and T. Perez. 2014. Fine-grained Plant Classification Using Convolutional Neural Networks for Feature Extraction, In Working Notes of CLEF 2014 Conference, Eds. L. Cappellato, N. Ferro, M. Halvey, and W. Kraaij, pp. 756–762. Sheffield, The United Kingdom.

Szczypiński, P. M., A. Klepaczko, and P. Zapotoczny. 2015. Identifying barley varieties by computer vision. *Computers and Electronics in Agriculture* 110:1–8. doi:10.1016/j.compag.2014.09.016.

Szczypiński, P. M., and P. Zapotoczny. 2012. Computer vision algorithm for barley kernel identification, orientation estimation and surface structure assessment. *Computers and Electronics in Agriculture* 87:32–38. doi:10.1016/j.compag.2012.05.014.

Xiong, C., L. Liu, X. Zhao, S. Yan, and T. K. Kim. 2016. Convolutional fusion network for face verification in the wild. *IEEE Transactions on Circuits and Systems for Video Technology* 26 (3):517–528. doi:10.1109/TCSVT.2015.2406191.

Xu, L., A. Krzyzak, and C. Y. Suen. 1992. Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE Transactions on Systems, Man, and Cybernetics* 22 (3):418–435. doi:10.1109/21.155943.

Ye, H., Z. Wu, R.-W. Zhao, X. Wang, Y.-G. Jiang, and X. Xue, 2015. Evaluating two-stream cnn for video classification, in: Proceedings of the 5th ACM on International Conference on Multimedia Retrieval. ACM, pp. 435–42. doi:10.1145/2671188.2749406.

Zapotoczny, P., M. Zielinska, and Z. Nita. 2008. Application of image analysis for the varietal classification of barley. *Journal of Cereal Science* 48 (1):104–110. doi:10.1016/j.jcs.2007.08.006.