Comparison of Light-Weight Multi-Scale CNNs for Texture Regression in Agricultural Context

Tilo Strutz and Alexander Leipnitz
Deutsche Telekom, Leipzig University of Telecommunications (HfTL), Institute of Communications Engineering
Gustav-Freytag-Str. 43–45, 04277 Leipzig, Germany

***Abstract*—While texture classification has a long history in image preprocessing tasks, its application in agriculture has only recently gained attention in the context of digital farming. The usage of camera drones allows the inspection of fields with a view from above. This paper proposes a method for the patch-based classification of different and basic ground regions using texture regression with convolutional neural networks. Two shallow multi-scale architectures are compared, which differ in the re-use of feature maps. It can be shown that a light-weight network is able to classify the textures with high accuracy. The performance is checked using the standard data set KTH-TIPS2b. The classification information can be effectively utilised for semantic image segmentation.**

***Index Terms*—texture classification, convolutional neural network, multi-scale, aerial images, image segmentation**

I. Introduction

With the availability of affordable drones equipped with high quality cameras, the usage of aerial images has found many applications. In the context of digital farming, these images can be analysed with different aims as for example, the discrimination between different crops [1] (combination of raw data and colour histograms) or the weed-plant detection in rice fields [2]. The optical information can be combined with images recorded in the near-infrared (NIR) band [3] for mound detection and counting. Also in [4] and [5], different spectral bands are used either for the discrimination of maize and soil for crop estimation or for the crop-type classification.

In conventional image processing, texture classification is based on features that are a result of image analysis in frequency bands (Gabor transform, Wavelet transform) and its spatial domain counterparts (e.g. Law's masks) or it utilises statistical measures based on grey-level co-occurrence matrices (GLCM) [2], [5]. Another famous approach is based on local binary patterns, e.g. [3], and its derivatives as shown in [6], to enumerate only the most prominent methods. As these methods operate in the grey-level domain, special preprocessing of colour images is needed. These feature-generating methods are combined with classification methods like support-vector machines (SVM) [4], random forest [5], or nearest neighbours [7]. An excellent overview of methods in texture analysis is given in [8].

Early attempts to exploit texture in agriculture relied on synthetic aperture radar (SAR) as, for example, in [9]. Guijarro *et al* reduced texture to colour information for segmenting pictures from an optical camera into plant, soil, and sky regions [10], while Campos *et al* aimed at the discrimination of plants, soil, and disturbing objects based on a camera mounted at a tractor. They used statistical feature including those derived from GLCM and SVM classification [11]. GLCMs are also investigated by [12] and performed worse than histograms of gradients (HoG).

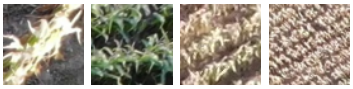

Fig. 1. Maize field textures at different scales and stages of maturity

In recent years, convolutional neural networks (CNNs) became more popular for texture classification, not only because they can combine feature extraction and classification in one system. Andrearczyk *et al*, for example, enriched a standard CNN with a special energy layer that collects activation output while ignoring its locality [13]. The truncation of CNNs at the level of convolutional layers (i.e. at least ignoring the fully connected layers) is considered in [14]. The resulting output serves as basis for conventional statistical features. The combination of handcrafted features with a CNN-based preprocessing step has also been investigated in [15]. Enforcing special learning directions, [16] incorporated locality and sparsity constraints into the CNN. The investigations also showed that CNN-based classification methods perform better than those purely based on handcrafted features do.

One of the major problems in the discrimination of texture is its scale. At further reflection, the scale problem is twofold:

a) The characterisation of a textures typically requires at least two scales, since different textures may look similar to each other when analysed at only a single scale.

b) The observation of a texture can occur at different distances between object and camera influencing the textural appearance, **Figure 1**. Camera settings like zoom and field of view have the same effect. As this scaling influences the sampling frequency with respect to the real world, the taken pictures contain different information about the observed objects.

Extreme scale variations may even cross borders between different descriptions, like 'leaf' → 'tree' → 'forest', which needs special treatment [17].

This paper investigates shallow CNNs in application to the discrimination of different textures (including colour) that can be observed from the air in agricultural context, that is, we concern textures that appear in the wild and clutter. In contrast to [12] and [18], we stick with basic classes of ground objects or regions and do not distinguish several growth stages of different plants.

The decision in favour to CNN has several reasons. On the one hand, the feature extraction based on convolutional

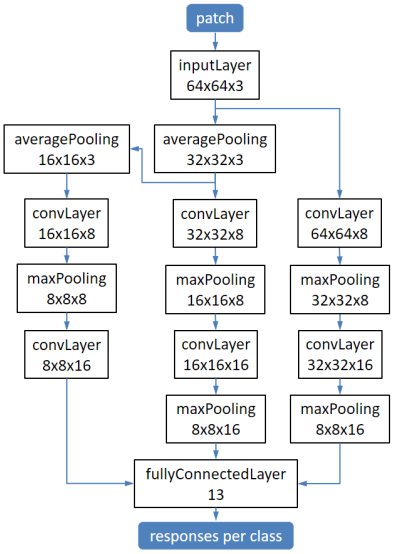

Fig. 2. Parallel multi-scale architecture of regression network

layers can automatically deal with colour information which is significant in agricultural context; on the other hand, convolutional neural networks have been proven to mostly adapt to textures and not to shapes in the context of object detection and classification [19]. Therefore, CNNs are ideal candidates when looking for an approach to successful texture classification.

This paper concentrates on medium scale variations and tackles the problem with multi-path CNN architectures that operated on image patches of size 64×64. Patch-based processing allows faster training of the CNN and requires less image data since many patches can be extracted from a single image. Additionally, it is suitable for arbitrary image sizes; no scaling of the input images to the size of the input layer is needed. The complexity of the proposed CNNs is kept low supporting cheap hardware. The performance of the net architectures has been tested using a thirteen-class data set 'Agri-13' and is validated using the KTH-TIPS2b data set [20].

One significant characteristic of the chosen CNN architectures is the output regression layer. In contrast to classification layers, it provides not the final class label but scores which can be interpreted as reliability measure. By doing so, more information of the texture analysis is kept and can supportively be used in the post processing. As an example application, we provide promising results for semantic image segmentation.

II. Design of Convolutional Networks

A. Architecture

The first proposed multi-path architecture is depicted in **Figure 2**. It consists of three paths with two convolutional layers and a max-pooling layer in between. This ensures that features on two scales of the input texture are analysed in each path. The three paths process the input patch at three different resolutions (provided by average-pooling layers). This tackles the problem of varying scales at object observation, as described in Section I.

TABLE I
Number of Adjustable Parameters

layer	number of weights	
conv $16 \times 16 \times 8$	$(3 \times 3 \times 3 + 1) \times 8$	224
conv $32 \times 32 \times 8$	$(3 \times 3 \times 3 + 1) \times 8$	224
conv $64 \times 64 \times 8$	$(3 \times 3 \times 3 + 1) \times 8$	224
conv $8 \times 8 \times 16$	$(3 \times 3 \times 8 + 1) \times 16$	1168
conv $16 \times 16 \times 16$	$(3 \times 3 \times 8 + 1) \times 16$	1168
conv $32 \times 32 \times 16$	$(3 \times 3 \times 8 + 1) \times 16$	1168
fully connected	$(8 \times 8 \times 16 \cdot 3 + 1) \times 13$	39949
	total:	44125

Figure 2 shows all layers and the size ($height \times width \times depth$) of their output. All convolutional layers use receptive fields of $3 \times 3 \times D$, with D being the depth (number of feature maps), which have been created by the previous layer. The stride and padding settings are chosen such that output feature maps have the same size as the input. A leaky ReLU layer follows each the convolutional layers. The multiplier for negative input values was chosen to be 0.01.

The pooling layers use a stride of $[2, 2]$ and a pool size of 2×2 for non-overlapping pooling. The only exception is the last max-pooling layer in the path on the right. It must reduce the size of feature maps from 32×32 down to 8×8. So, it has a stride of $[4, 4]$ and a pool size of 4×4.

The results of all three paths are fully connected with an output layer consisting of $N_{\text{classes}} = 13$ neurons. This corresponds to regression where the target vector consists of $N_{\text{classes}} - 1$ zero elements and a single '1' at the position of the correct class. Similar to the 'soft decision' procedure used in forward-error control coding, this set-up keeps information about the reliability of the class decision and allows a more fine-granular interpretation of the result. The output of a single label corresponds to 'hard decision' and causes loss of valuable information.

Tab. I lists the number of adjustable parameters. The notation for the convolution layers is $(H \times W \times D + B) \times F$ with $H \ldots$ height, $W \ldots$ width, $D \ldots$ depth, $B \ldots$ bias, and $F \ldots$ number of filters. The fully connected layer has $8 \times 8 \times 16 \cdot 3 = 3072$ input nodes[1] and thirteen output neurons. With only six convolutional layers and mere 44125 adjustable parameters, this architecture can be considered as light-weight.

It should be mentioned that the input layer normalises the data to the range 0...1 by dividing all components of all pixels by 255.

The parallel-paths architecture is compared to a more sequential design that is similar to standard CNNs like AlexNet [21] with additional short-cuts, **Figure 3**. The most right path is identical to the architecture in Figure 2. Each of the other paths uses the result of processing steps of another path. As the number of convolution layers and the numbers of their filters are identical, the complexity of both networks also are the same enabling a fair comparison.

III. Investigations

Data preparation, training and testing of the CNNs have been carried out using MATLAB® R2019.

[1]This number defines the complexity per class when the number of classes goes to infinite.

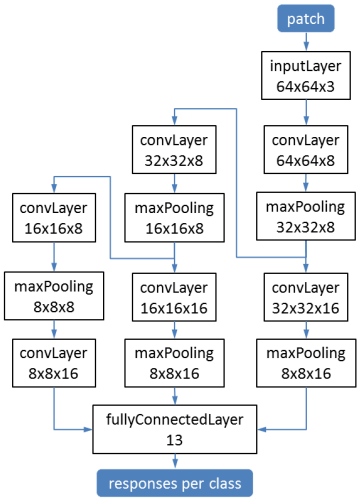

Fig. 3. Sequential re-use of layers making the multi-scale network deeper

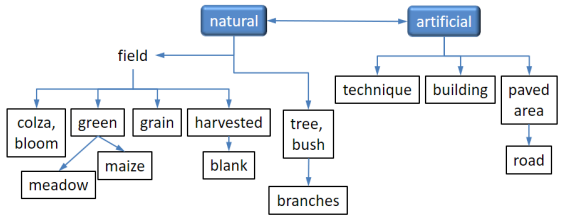

Fig. 4. Considered classes and their relation

A. Data collection

Aerial pictures have been taken from spring to early winter with two different drones: 'DJI Mavic 2 Enterprise' and 'DJI Mavic Pro'; the image resolution is 3840×2160 pixels. The chosen camera orientation angle was about $90°$ for top-down view introducing only slight perspective distortions. The flight altitude varied between about 15 and 50 metres.

The taken pictures have been downscaled by factor 0.5 to reduce the influence of possible compression artefacts before they have been segmented into patches of 64×64 pixels. Additionally, the patches have randomly been rotated by multiples of $90°$ reducing the presence of dominating orientations. Then, the patches have been sorted into thirteen classes as shown in **Figure 4**. Patches with mixed content were ignored. The patch size is a good compromise between sufficient coverage and classification resolution.

The selection of classes reflect the most important and most probable cases in the contemplated scenario. An extension could include (partially more specific) classes like: water, snow, hedge, animal, person, fence, tractor etc.

Typical appearances of each class have been selected, which automatically led to categories with some visual overlap like 'road' and 'paved area', 'green_field' and 'meadow', or fields that are blank or harvested. Class 'branches' could be regarded as sub-class of 'tree_bush', which itself contains two different plants because trees and bushes look very similar when observed from the top. Since we did not record depth

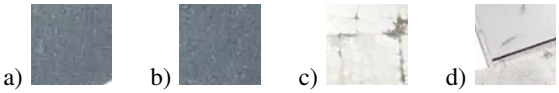

Fig. 5. Examples for similar appearances of different classes: a) road, b)+c) paved area, d) building

Fig. 6. Examples for different appearances within the same class 'meadow'

information, the height of the plants remains unknown and the discrimination between bushes and trees is almost impossible on textural basis.

Gathering pictures for the class 'technique' was most difficult because tractors and other agricultural machinery can be met only at certain times and often in only limited variety. It was necessary to augment this class based on royalty-free images from the internet. For each class, more than 3000 patches have been collected.

The data set does not only show high inter-class similarity, **Figure 5**a)+b) and c)+d), but also a high intra-class diversity as can be seen based on the example patches in **Figure 5**b)+c) and **Figure 6**.

The two proposed CNN architectures have additionally been tested with the KTH-TIPS2b data set from [20]. It comprises eleven classes. The test images have been segmented into 64×64 patches in the same manner as the agricultural pictures. As the typical image size is 200×200 in this data set, which is not a multiple of 64, all images also have been rotated by $180°$ leading to new and shifted segments. The number of generated patches per class varies from 5960 to 7776 depending of the original image sizes. Note that the diversity of textures has been increased by the random rotation of the patches. The original data of KTH-TIPS2-b shows the same textural orientation in some classes.

B. Hyperparameter

The data sets have been separated into training data (60%), validation data (20%), and test data (20%). Along with the default settings in Matlab, following hyperparameters have been selected for the training procedure: stochastic-gradient-descent optimizer with momentum (SGDM), a learning rate of 0.01, a mini-batch size of 128, and shuffling of the input data at the beginning of the training. After multiples of 16 epochs, the network is applied to the validation data set. The training stops if the validation result does not improve after ten trials (i.e. after 160 epochs).

Alternatively, the experiments have been carried out using the Adam optimiser. However, the training-loss curve showed spikes and the training got stuck in local minima. For this reason, Adam optimizer is not considered in the results section.

IV. Results

In total, six experiments have been carried out. The two architectures have been investigated in combination with three

TABLE II
ACCURACY FOR DIFFERENT TESTS AND HYPERPARAMETER

architecture\ data	Agri-13	KTH-TIPS2b	merged
parallel, sgdm	88.7%	98.4%	85.3%
sequential, sgdm	87.0%	97.6%	83.8%

different data sets: Agri-13, KTH-TIPS2b, and a merged data set (Agri-13 + KTH-TIPS2b).

A. Training, validation

The networks have been trained from the scratch, provided with a balanced number patches, and without further fine-tuning. In general, the loss rapidly drops in the beginning of the training and then converges to a minimum.

In order to check the capacity of the networks, we increased the complexity of the classification task by merging the two data sets to a single one comprising 24 classes.

The trainings have been performed on a single GPU and last about one to two hours for the Agri-13 data set.

B. Testing

After training, the CNNs have been applied to the test-set data. The best results of several trials of the different experiments are listed in **Table II**. The accuracy is the number of correct classified patches (after hard decision in favour of the class with the highest regression score) divided by the total number of analysed patches in percent. The values are between 87% and 89% for the Agri-13 data set. This is a good result if one bears in mind that different classes share very similar patches. Most misclassifications occur between 'harvested_field' vs 'blank_field' and 'road' vs 'paved area'. Also 'technique' and 'building' show some overlap.

The results for the KTH-TIPS2b data set are distinctly better. This could be expected, because the intra-class similarity is much higher and the inter-class overlap much lower.

Experiments with the merged data set reveal that the capacity of such a simple multi-scale multi-path architecture is close to its limits. The accuracy is dropped down to about 84%, also caused by additional false classification between classes of the two original data sets.

There seems to be no significant performance difference between the two different architectures.

V. SEMANTIC IMAGE SEGMENTATION

Semantic image segmentation (see for example [22]) is one possible application of patch-based texture classification. As it is too costly to apply the network to patches at all possible pixel positions, an effective method is required to merge the classification results from neighbouring non-overlapping patches. The scores of all classes are piece-wise linearly averaged between the centres of neighbouring patches. This supports classes with high scores and suppresses other classes. The resolution and reliability is further enhanced by applying the analysis to versions of the input image at two different scales (factor 0.5). For each pixel, the maximum score of both versions is used. This additionally increases the scale variability of the system. **Figure 7** shows the results for one sample image. The applied assignment of colours and class labels in Fig 7b) is depicted in **Figure 8**.

If the classification score is either to low (< 0.3) or the score of the dominant class is not at least ten percent higher than the second best, the pixel's class is annotated as 'unknown'. This happens typically at region borders. A special effect concerns the class 'technique'. It contains many patches with strong edges leading to the effect that regions with similar edges also are classified as 'technique', as can be seen in Fig, 7b) for building boundaries or the person on the road. The paved area between the two buildings is falsely identified as road and some non-existing bushes are indicated.

VI. SUMMARY

The paper has shown that texture classification in the agricultural context is possible based on light-weight convolutional neural networks (CNN). The proposed architectures aim at analysing the input image at different resolutions combating the well-known scale problem in texture recognition.

The proposed CNNs operate on patch basis supporting the processing of arbitrary image sizes without the need of re-scaling. They can be trained from the scratch in moderate time.

The performance of the two investigated architectures is about the same. Albeit the more sequential CNN is deeper on average, this advantage seems to be alleviated by the shared convolutional layers. Probably, the robustness in the wild could be enhanced by using colour balancing before presenting the images to the CNN as proposed in [23].

The application of the proposed texture-classification method to image segmentation using a voting scheme for the most probable texture achieves already very good results. However, semantic relations that are larger than the patches cannot be incorporated, which leads to fuzzy region boundaries. The fusion with supplemental image analysis methods would be reasonable for enhancing the segmentation results.

Supporting reproducible research, the software used for the investigations can be downloaded from [24].

VII. ACKNOWLEDGEMENTS

The authors would like to thank Jan Heine for the provision of one part of the data set. The work has financially be supported by the Federal Ministry of Education and Research of Germany in the framework of the Era.Net HARMONIC project (project number 01DJ18011).

REFERENCES

[1] J. Rebetez, H. F. Satizabal, M. Mota, D. Noll, L. Büchi, M. Wendling, B. Cannelle, A. Perez-Uribe, and S. Burgos, "Augmenting a convolutional neural network with local histograms - a case study in crop classification from high-resolution uav imagery," in *Proceedings of ESANN 2016*, Bruges, Belgium, April 2016.

[2] O. Barrero, D. Rojas, C. Gonzalez, and S. Perdomo, "Weed detection in rice fields using aerial images and neural networks," in *Proceedings of STSIVA 2016*, Bucaramanga, Colombia, August 2016.

[3] W. Bouachir, K. E. Ihou, H.-E. Gueziri, N. Bouguila, and N. Belanger, "Computer vision system for automatic counting of planting microsites using UAV imagery," *IEEE Access*, vol. 7, pp. 82 491 – 82 500, June 2019.

[4] O. Hall, S. Dahlin, H. Marstorp, M. F. A. Bustos, I. Öborn, and M. Jirström, "Classification of maize in complex smallholder farming systems using UAV imagery," *MDPI drones*, vol. 2, no. 22, pp. 1 – 8, 2018.

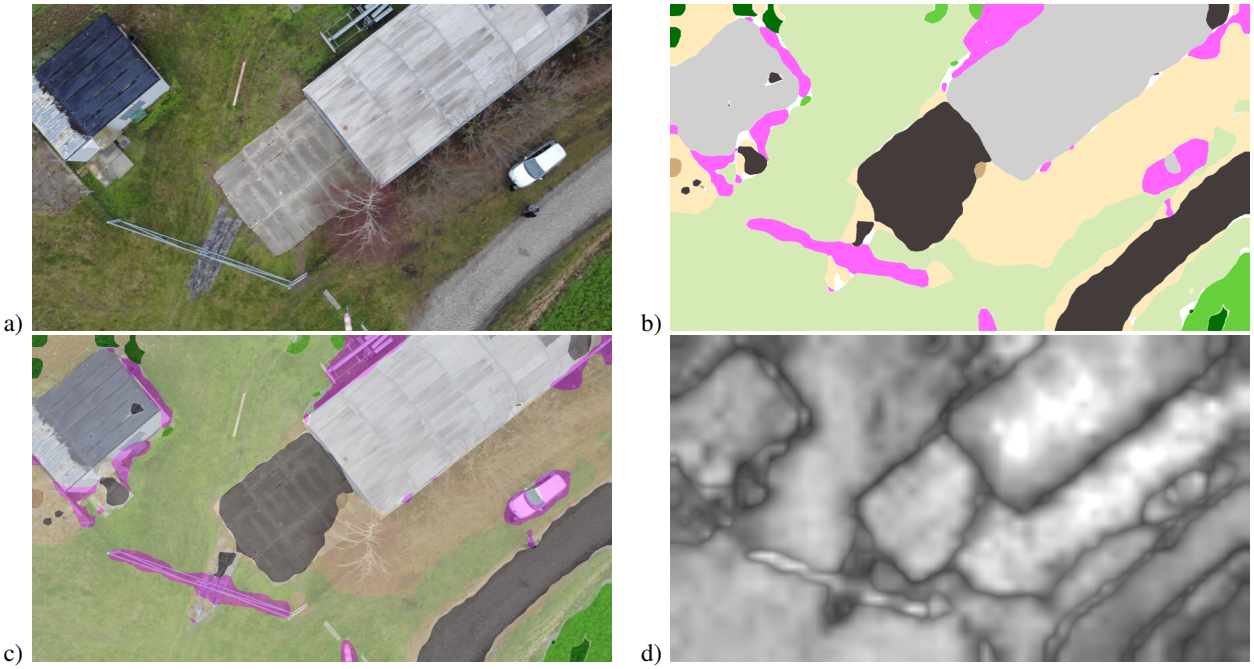

Fig. 7. Example of image segmentation based on patch classification (parallel architecture, sgdm): a) original, b) segmented, c) overlay of a)+b), d) scores of classification in the range from 0 (black) to 1 (white, highest reliability)

0: unknown	5: colzaBloom	10: building
1: branches	6: meadow	11: pavedArea
2: harvested_field	7: maize_field	12: road
3: blank_field	8: green_field	13: technique
4: grain_field	9: tree_bush	

Fig. 8. Assignment of class numbers, labels and colours

[5] H. Zhang, Q. Li, J. Liu, J. Shang, X. Du, H. McNairn, C. Champagne, T. Dong, and M. Liu, "Image classification using RapidEye data:integration of spectral and textual features in a random forest classifier," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 12, pp. 5334 – 5349, December 2017.

[6] L. Liu, S. Lao, P. W. Fieguth, Y. Guo, X. Wang, and M. Pietikäinen, "Median robust extended local binary pattern for texture classification," *IEEE Trans. on Image Proc.*, vol. 25, no. 3, pp. 3071 – 3084, July 2016.

[7] Y. Dong, T. Wang, C. Yang, L. Zheng, B. Song, L. Wang, and M. Jin, "Locally directional and extremal pattern for texture classification," *IEEE Access*, vol. 7, pp. 87 931 – 87 942, July 2019.

[8] L. Liu, J. Chen, P. Fieguth, G. Zhao, R. Chellappa, and M. Pietikäinen, "From BoW to CNN: Two decades of texture representation for texture classification," *International Journal of Computer Vision*, vol. 127, no. 1, pp. 74 – 109, 2019.

[9] P. Treitz, O. Filho, P. Howarth, and E. Soulis, "Textural processing of multi-polarization SAR for agricultural crop classification," in *Proceedings of IGARSS 1996*, Lincoln, NE, USA, May 1996.

[10] M. Guijarro, G. Pajares, I. Riomoros, P. J. Herrera, X. P. Burgos-Artizzue, and A. Ribeiro, "Automatic segmentation of relevant textures in agricultural images," *Computers and Electronics in Agriculture*, vol. 75, no. 1, pp. 75 – 83, January 2011.

[11] Y. Campos, H. Sossa, and G. Pajares, "Comparative analysis of texture descriptors in maize fields with plants, soil and object discrimination," *Precision Agriculture*, vol. 18, no. 5, pp. 717 – 735, October 2017.

[12] H. Yalcin, "Phenology monitoring of agricultural plants using texture analysis," in *Proceedings of Fourth International Conference on Agro-Geoinformatics, 2015*, Istanbul, Turkey, July 2015.

[13] V. Andrearczyk and P. F. Whelan, "Using filter banks in convolutional neural networks for texture classification," *Pattern Recognition Letters*, vol. 84, pp. 63 – 69, December 2016.

[14] M. Cimpoi, S. Maji, I. Kokkinos, and A. Vedaldi, "Deep filter banks for texture recognition, description,and segmentation," *International Journal of Computer Vision*, no. 5, Januar 2016.

[15] M.-X. Bastidas-Rodriguez, F.-A. Prieto-Ortiz, and L. F. Polania, "A textural deep neural network combined with handcrafted features for mechanical failure classification," in *Proceedings of IEEE International Conference on Industrial Technology (ICIT) 2019*, Melbourne, Australia, February 2019.

[16] X. Bu, Y. Wu, Z. Gao, and Y. Jia, "Deep convolutional network with locality and sparsity constraints for texture classification," *Pattern Recognition*, vol. 91, pp. 34 – 46, July 2019.

[17] L. Liu, J. Chen, G. Zhao, P. W. Fieguth, X. Chen, and M. Pietikäinen, "Texture classification in extreme scale variations using GANet," *IEEE Trans. on Image Proc.*, vol. 28, no. 8, pp. 3910 – 3922, August 2019.

[18] F. Gulac and U. Bayazit, "Plant and phenology recognition from field images using texture and color features," in *Proc. of Innovations in Intelligent Systems and Applications (INISTA), 2018*, Thessaloniki, Greece, July 2018.

[19] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, "ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness," in *Proceedings of ICLR 2019*, New Orleans, USA, May 2019.

[20] P. Mallikarjuna, A. Targhi, M. Fritz, E. Hayman, B. Caputo, and J.-O. Eklundh, "The KTH-TIPS2 database," in *Proceedings of ICCV 2006*, July 2006.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[22] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3431 – 3440.

[23] S. Bianco, C. Cusano, P. Napoletano, and R. Schettini, "Improving cnn-based texture classification by color balancing," *J. Imaging*, vol. 3, p. 33, 2017.

[24] T. Strutz, "Texture classification and image segmentation," Available: http://www1.hft-leipzig.de/strutz/Papers/TextureClassif-resources/, accessed: August 20, 2020 [Online].