A New Assessment of Convolutional Neural Networks for Texture Directionality Detection

Marcin Kociołek[1] and Antonio Cardone[2]

[1] Lodz University of Technology Institute of Electronics Al. Politechniki 10, 93-590 Lodz, Poland
marcin.kociolek@p.lodz.pl,
WWW home page: http://https://ie.p.lodz.pl
[2] National Institute of Technology, Software and Systems Division, 100 Bureau Drive Gaithersburg, MD 20899, USA

Abstract. Image texture analysis is ubiquitous as it finds applications in many scientific fields of interest, including biomedical and material science. The detection of meaningful texture properties such as directionality remains a challenging task, due to the complexity of texture. We build upon our past work on the design of convolutional neural networks (CNNs) for texture directionality detection. The CNNs are trained on a library of synthetic textures with known directionality and varying perturbation levels. The present effort focuses on enhancing the training data through a new perturbation procedure and a more diverse set of synthetic textures. We study the performance of new CNN architectures, such as grouped CNNs, on the enhanced synthetic texture library. The results yield novel insight into CNN-based texture directionality detection. Shallow and grouped CNNs show better performance than deep CNNs, unlike previously. We discuss this performance shift and its implications, and suggest possible future work directions.

Keywords: directionality detection, texture analysis, image processing

1 Introduction

Image texture analysis is challenging due to the complexity of texture local patterns, or textons [1]. Texture directionality is an important property whose study finds applications in several domains. In the past we proposed the use of convolutional neural networks (CNN) trained on synthetic textures for texture directionality detection [2]. The synthetic textures consist of patterns with known directionality and increasing levels of perturbations (i.e., blur and additive noise) to mimic real-life texture images (Fig. 1). The resulting library consists of textures whose directionality is discernable by a human observer.

The previously designed CNNs consisting of one convolutional layer (henceforth shallow CNNs) and of three convolutional layers (henceforth deep CNNs). The CNNs were trained, validated and tested on the synthetic texture library. Both shallow (SN1) and deep (DN2) CNNs were found to perform better than

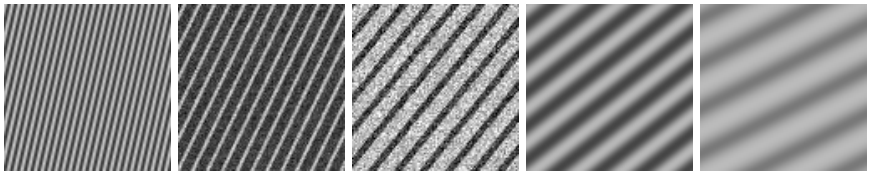

Fig. 1. Instances of synthetic texture images from our synthetic texture library

or comparably to state-of-the-art techniques. Qualitative assessment suggested that the CNNs trained on synthetic textures nicely generalize to real-life images [2].

In this paper we improve the training data by eliminating aliasing effects [2] and by using synthetic texture images of different sizes. Texture is a local property of images whose patterns typically have different sizes, and hence multi-size training textures more closely mimic real-life textures. Furthermore, we get additional insight by testing previously and newly designed CNNs on the enhanced training data, and by using additional activation functions [2]. Most of the new CNNs consist of grouped CNNs, in which the convolutions are performed independently for each group at any given layer. Grouped CNNs reduce the number of computations and enable a better representations of the data.

The paper is organized as follows. In Section 2, the related literature is surveyed. We describe the enhanced training data and the CNN architectures in Section 3. The results are presented in Section 4, and our conclusions and future work directions are discussed in Section 5.

2 Related Work

Texture directionality analysis plays a fundamental role in fields such as image retrieval and classification [3–5]. In the biomedical field, therapeutic approaches often rely on the characterization of texture directionality, as in placebo-controlled trials [6]. In material science and engineering, texture directionality is correlated to material properties [7]. In astronomy, high-throughput texture directionality analysis has been applied to solar images [8].

Texture directionality detection approaches rely on computational entities such as the Fourier transform [9] and the local gradient orientation (LGO) [10](implemented in the Fiji/ImageJ software [11]), the Radon transform [12], and the gray level co-occurrence matrix (GLCM) [13]. The design of the underlying models can be challenging, as the parameters need to be customized for the targeted data. Many models involve heavy computations and are not viable for high-throughput efforts.

Supervised machine learning can circumvent challenging model designs. We have tested CNNs trained on synthetic textures for directionality detection, and have identified shallow and deep CNNs with accuracy comparable to the state-

of-the-art and less computational load [2]. Still, the design space of CNNs needs to be further explored.

Grouped CNNs are potentially effective for image analysis. Their convolutional layers are organized in groups with independently performed convolutions. They were introduced first in AlexNet [14], evidence shows improved classification accuracy [15]. One reason could be that grouped CNNs introduce a new dimension across the grouped convolutions, cardinality, with size proportional to the network accuracy [15].

A related topic is the equivalence of a cascade of smaller filters (e.g., smaller convolutions in a grouped CNN) to a larger filter (e.g., larger convolution in a shallow CNN). Given a large filter, an equivalent cascade of smaller filters has always fewer parameters, hence it can be found only under specific conditions [16]. As such equivalence is possible, it is worth studying grouped CNNs in this context.

3 Materials and Methods

The work presented in this paper is two-fold: we improve the training data used in our previous effort [2] and we test previously and newly designed CNN architectures on the enhanced training data, with additional activation functions.

3.1 Enhanced Training Data

The quality of training data is of paramount importance for the training of CNNs. In this work, we build an enhanced version of the synthetic texture image library.

First, we address the aliasing effects caused by averaging filters [2] by employing Gaussian filters instead [17]. Different blurring levels are obtained by varying the standard deviation σ within a range. The range was selected empirically by visual inspection by ensuring no aliasing effects are present at any given bar period and thickness combination. Table 1 shows the corresponding maximum Gaussian standard deviation σ values.

Table 1. Bar periods, bar sizes and maximum σ for the Gaussian blur used in this paper

bar period	4	4	4	6	6	6	6	6	8	8	8	12	12	12
bar size	1	2	3	1	2	3	4	5	2	4	6	2	4	8
blur maximum σ	0	0	0	0	1	1	1	0	2	2	2	4	4	4
bar period	12	16	16	16	20	20	20	24	24	24	30	30	30	30
bar size	10	4	8	12	5	10	15	6	12	18	6	12	18	24
blur maximum σ	4	5	5	5	6	6	6	6	7	7	8	8	8	8

The rationale behind Gaussian filters is that they act as low-pass filters, as the Gaussian function yields a smoother impulse response. Instead, averaging filters

cause sharp transitions and they act as step filters. Gaussian filters mitigate the spatial anisotropy associated with the filter shape, in our case a square, since the pixels away from the center are weighted by a bell-curve. Gaussian filters are less likely to lead to the same degree of aliasing effects as averaging filters (Fig. 2).

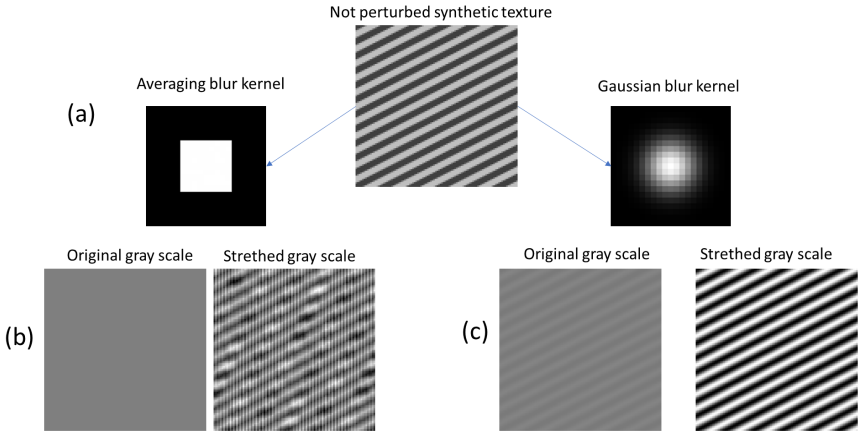

Fig. 2. a) Unperturbed synthetic texture, averaging (left) and Gaussian (right) perturbing kernels; upon perturbation using the kernels in (a), synthetic textures with (b) and without (c) aliasing effect

Second, we improve the synthetic texture library by diversifying it. As texture is a local image property associated to repeating local patterns, automated texture analysis needs to deal with texture patterns of different sizes. Hence, the synthetic textures used in this paper span four different sizes, 38×38, 46×46, 54×54, 62×62. This more closely mimics real-life textures, and the CNNs trained on them will have enhanced generalization capabilities. Overall, the training data consist of 1,490,760 texture images, the validation data of 745,380 texture images, and the test data of 745,380 texture images. Another improvement results from the broader range of bar period and thickness combinations used.

3.2 Novel CNN Architectures and Additional Activation Functions

We study twelve CNNs, some previously designed [2], whose architecture is shown in Fig. 3.

The CNNs were implemented using the TensorFlow library (Keras module) with a Python interface (Python version 3.10.5, TensorFlow version 2.7.1) [18]. All CNN architectures contain a global max pooling layer to enable a wide range of input image sizes, as opposed to only one image size. CNNs consist of one or several convolutional layers, generally followed by a pooling layer. Each

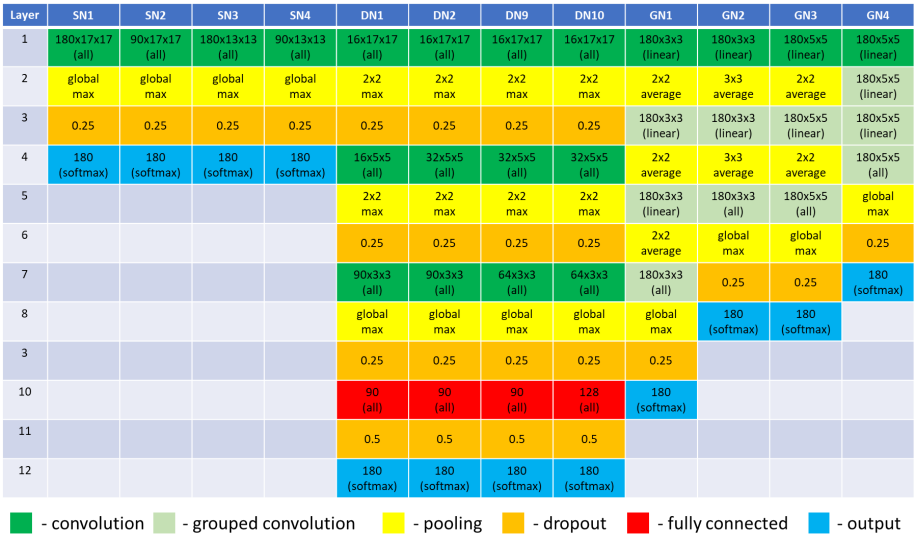

Fig. 3. Detailed schematic of all the CNN architectures used in this paper. For a given convolutional or fully connected layer, the specific activation function used is reported in parenthesis ("all" implies all activation functions were tested).

convolutional layer is effectively a set of feature extraction filters. Note that, as in our previous work, we consistently employ convolutions with no padding [2]. The reason is that padding can cause artifacts at the image edge, likely leading to the detection of wrong texture directions. In the absence of padding, a sequence of convolutions will always result in a feature map of smaller size. Quantitatively, given a convolutional layer consisting of filters of size $(k \times l)$ without padding, an input image of size $(n \times m)$ will produce an output image of size $((n-k+1) \times (m-l+1))$. In addition, a pooling operation of dimensions $(o \times p)$ on an image of dimensions $(n \times m)$ will produce an image of dimensions $((n/o) \times (m/p))$. A sequence of K convolutions operations can be formally defined as follows.

$$I_{out}[\overline{n}, \overline{m}] = F_K * F_{K-1} * \ldots * F_2 * F_1 * I_{out}[n, m]. \quad (1)$$

In Equation 1, K is the number of convolutions (with or without pooling layers), and I_{in} and I_{out} are the input and output images with respective sizes $(n \times m)$ and $(\overline{n} \times \overline{m})$. As pointed out, $n > \overline{n}$ and $m > \overline{m}$. Based on Equation 1, two sequences of convolutions are size-equivalent if, given same size input images, the output images are also of the same size. The CNNs described in this paper always utilize a global max pooling layer after the last convolutional one. This enables processing input images of variable size. Since the minimum allowed image size for a global max pooling layer is 1×1, the minimum input image size for our CNNs is constrained accordingly. Hence, size-equivalent CNNs are also defined as having the same minimum allowed input image size. The expectation

is that size-equivalent CNNs can be trained to have similar prediction performances, hence achieving the equivalence between cascades of filters and single large filters discussed in the Introduction. Here, network GN1 is size-equivalent to the networks DN1 and DN2 from our previous work, and network GN4 is size-equivalent to SN1 [2].

The role of the activation functions is critical for the CNN performances. Depending on properties such as boundaries, symmetry, and linearity, given activation functions will work better with specific CNNs architectures. We tested five additional activation functions available with Tensorflow 2.7.1, not included in our previous work: linear (ln), exponential (ex), Swish (sw), Gaussian error linear unit (ge), and hard sigmoid (hs). The previously employed activation functions, still used here, are: exponential linear unit (el), rectified linear unit (re), scaled exponential linear unit (se), sigmoid (si), soft plus (sp), soft sign (ss), and hyperbolical tangent (th).

4 Results and Discussion

The CNNs discussed in the Materials and Methods section were trained, validated and tested with the enhanced synthetic texture library. As in our previous work [2], we use the following directionality prediction error, which considers the inherent 180° periodicity.

$$error = \arccos(cos(|\alpha - \beta|)) \tag{2}$$

In Equation (2) α is the true direction, and β is the predicted one.

Each CNN architecture was trained and validated in three replicates using Keras libraries and functions [19]. The training and validation curves show consistency across replicates. Good convergence behavior is evident, especially for the best-performing architectures, based on the categorical cross entropy (loss function) and RMSE. Subsequently, all CNNs architecture were tested. The metrics to evaluate CNN test performances are root mean square error (RMSE) and maximum error (MxE), both based on Equation 2. RMSE characterizes the average performance of a CNN, MxE reflects the presence of large errors, not acceptable in many applications. The tests yield novel insight into CNN performance, discussed as follows. First, our data clearly show major improvements in CNN performances when trained on the enhanced synthetic texture library. An instance is the performance of the best shallow CNN SN1 (see Table 2).

This is probably due to the absence of aliasing effects in the enhanced synthetic texture library, which improves the quality of the training and increases the generalization capability of the CNN. The presence of multi-size training data might also force the CNN to deal with more diverse texture patterns, which yields enhanced generalization capability.

The data also show that shallow CNNs have now a clearly better performance than deep CNNs, unlike our past results [2] (Table 3).

This is because the meaningful information about texture directionality probably lies in the lower-level features, which are collected in the first convolutional

Table 2. MxE and RMSE data for the best shallow network SN1, trained, validated and tested on both the old and the enhanced synthetic texture library.

Training, validation and test on the old synthetic texture library							
Activation	el	se	si	sp	re	ss	th
MxE	89	90	90	87	90	84	82
RMSE	2.1	2.2	3.1	3.2	3.6	4.3	6.3

Training, validation and test on the enhanced synthetic texture library												
Activation	ss	si	th	sp	se	ln	sw	el	ge	hs	ex	re
MxE	3	3	4	4	5	5	5	6	6	7	10	52
RMSE	0.28	0.29	0.32	0.34	0.32	0.34	0.40	0.32	0.37	0.33	0.74	0.74

Table 3. RMSE and MxE data for the best deep network DN2, trained, validated and tested on the enhanced synthetic texture library.

Activation	ss	si	th	sp	se	ln	sw	el	ge	hs	ex	re
MxE	3	3	4	4	5	5	5	6	6	7	10	52
RMSE	0.28	0.29	0.32	0.34	0.32	0.34	0.40	0.32	0.37	0.33	0.74	0.74

layer. Subsequent layers in the deep CNNs describe higher-level texture abstractions, such as relative location, which are less meaningful. This is more evident based on the data from the enhanced texture library, whereas it was not so clear from our previous data, due to the presence of aliasing effects. Additional tests (data not shown) were performed on deep CNNs with different number of filters (i.e., doubling in each subsequent layer), with no evidence of performance improvement. Tests on shallow CNNs with different sizes were also performed, but SN1 remains the best performing architecture. Additional tests focused on grouped CNNs, trained on the enhanced synthetic texture library. They span size-equivalent groups from the best shallow CNN, SN1, to the best deep CNN, DN2, from our previous work [2]. The best-performing grouped CNN is GN4 (Table 4).

Table 4. RMSE and MxE data for the grouped networks GN1, GN2 and GN4.

	Activation	ge	sw	el	se	sp	si	th	ss	hs	ex	re	ln
GN1	MxE	90	90	90	90	90	90	90	90	90	90	90	90
	RMSE	7.35	7.52	7.77	8.36	8.53	8.89	9.08	9.33	10.1	11.6	21.1	23.5
	Activation	ge	sw	el	se	si	hs	th	sp	ss	ex	re	ln
GN2	MxE	90	90	90	90	90	90	90	90	90	90	90	90
	RMSE	5.92	6.10	6.26	6.65	7.17	7.43	7.70	7.84	8.02	10.1	20.3	22.1
	Activation	se	el	ge	sw	si	ln	ex	sp	th	ss	hs	re
GN4	MxE	4	4	4	5	5	7	10	11	29	59	77	89
	RMSE	0.31	0.33	0.34	0.30	0.56	0.24	0.82	0.87	0.42	0.51	0.91	2.57

Note that, as far as size-equivalence, GN4 is closest to the best shallow network SN1. Both CNNs are probably close to the optimal size for texture directionality detection when employing our training data.

Notably, the best performing activation functions for the shallow network SN1 are different from our previous work (Table 2). Smooth rectifier activation functions such as ELU (el) and SELU (se) were previously reported to perform best [2]. With the enhanced synthetic texture library, mainly symmetric and bonded activation functions perform better, such as soft sign (ss), sigmoid (si), and hyperbolical tangent (th). This underscores the importance of the activation functions in the CNN performance.

Another important result is related to GN4, the best grouped CNN. In GN4, the linear activation function (ln), which is not bounded, is used in all convolutional layers except for the last one. In the last convolutional layer, smooth rectifier activation functions are used, such as SELU (se), as they yield the best performance. Overall, GN4 performance improves with non-bounded activation functions. This behavior is opposite to the best shallow CNN, SN1, where bounded activation functions such as SOFTSIGN (ss) perform best, probably due to their capability to counter the effect of outliers. While surprising at first, one explanation is that in GN4 non-bounded activation functions contribute with their properties to creating the conditions under which the cascade of smaller filters becomes equivalent to the unique larger filter of SN1 (cfr., Related Work section). Note that GN4, unlike the other grouped CNNs tested here, does not have any pooling operation between the convolutional layers. The absence of pooling operations could also contribute to the equivalence between the cascade of smaller filters of GN4 and the unique large filter of SN1.

5 Conclusions and Future Work

This effort has contributed to further elucidate the feasibility of CNNs trained on synthetic textures in the context of texture directionality detection. The employed synthetic textures closely resemble real-life ones, and hence the designed CNNs seem to nicely generalize to real-world images [2]. Overall, the best performing CNN is still the shallow SN1.

One important conclusion is that the quality of the training data is of fundamental importance. As evident from our previous results, the synthetic textures defined in the digital domain are associated with unwanted perturbation effects (i.e., aliasing effects), which may lead to the improper labeling of the training data. We mitigated adverse perturbation effects by employing enhanced perturbation procedures (i.e., Gaussian filters) and by ensuring no leftover aliasing effect is included in the training data. As a consequence, the performance of shallow CNNs clearly improved.

Another conclusion is that the better performance of shallow CNNs, when compared to deep CNNs, suggests that shallow CNNs probably enable the full characterization of texture directionality in the unique convolutional layer em-

ployed. Hence, there is probably no need to resort to deeper architectures for increased accuracy.

Lastly, our tests suggest that grouped CNNs, rather than deep CNNs, might be viable alternatives to shallow CNNs. This is possibly associated to the conceptual equivalence between cascades of smaller filters and large ones. Such equivalence seems to be suggested by the very similar performances of size-equivalent grouped and shallow CNNs (cfr., Materials and Methods).

The implementation of the described methods is available in an open Github repository [20].

Based on the above, one future work direction is to test additional grouped CNNs, consisting of a smaller or larger number of layers. As our intuition is that size-equivalent shallow and grouped CNNs might perform similarly in texture directionality detection, the new architectures should span as many size-equivalent shallow networks as feasible, including the best one, SN1. Another future work direction is related to the absence of pooling operations in the best-performing grouped network GN4. Whereas this might be evidence that such an absence contributes to the equivalence between the cascade of smaller filters in GN4 and the unique larger filter in shallow networks, it is still worth testing different CNN architectures with no pooling operations, including deep ones. This would provide a more complete picture of the performance of CNNs in texture directionality detection, possibly leading to further enhancements of the proposed approach.

Disclaimer

Commercial products are identified in this document in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the products identified are necessarily the best available for the purpose.

References

1. B. Julesz, E. N. Gilbert, L. A. Shepp, and H. L. Frisch, "Inability of humans to discriminate between visual textures that agree in second order statistics: revisited," Perception, vol. 2, no. 4, pp. 391–405, Jun. 1973, doi: 10.1068/p020391.
2. 2. M. Kociołek, M. Kozłowski, and A. Cardone, "A Convolutional Neural Networks-Based Approach for Texture Directionality Detection," Sensors 2022, Vol. 22, Page 562, vol. 22, no. 2, p. 562, Jan. 2022, doi: 10.3390/S22020562.
3. P. Hassekar and R. R. Sawant, "Experimental analysis of perceptual based texture features for image retrieval," Proceedings - 2015 International Conference on Communication, Information and Computing Technology, ICCICT 2015, pp. 1–6, 2015, doi: 10.1109/ICCICT.2015.7045666.
4. X. Lin, L. Ye, W. Zhong, and Q. Zhang, "Directionality-based modified coefficient scanning for image coding," Proceedings - 2015 8th International Congress on Image and Signal Processing, CISP 2015, vol. 1, no. CISP, pp. 194–198, 2016, doi: 10.1109/CISP.2015.7407874.

5. M. Maskey and T. S. Newman, "On measuring and employing texture directionality for image classification," Pattern Analysis and Applications, Jul. 2021, doi: 10.1007/s10044-021-01013-8.
6. M. Lin et al., "Longitudinal assessment of carotid plaque texture in three-dimensional ultra-sound images based on semi-supervised graph-based dimensionality reduction and feature selection," Comput Biol Med, vol. 116, p. 103586, Jan. 2020, doi: 10.1016/J.COMPBIOMED.2019.103586.
7. V. Edachery, S. R, and S. v. Kailas, "Influence of surface texture directionality and roughness on wettability, sliding angle, contact angle hysteresis, and lubricant entrapment capability," Tribol Int, vol. 158, p. 106932, Jun. 2021, doi: 10.1016/J.TRIBOINT.2021.106932.
8. D. J. Kempton, A. Ahmadzadeh, M. A. Schuh, and R. A. Angryk, "Improving the functionality of tamura directionality on solar images," Proceedings - 2017 IEEE Inter-national Conference on Big Data, Big Data 2017, vol. 2018-Janua, pp. 2518–2526, 2017, doi: 10.1109/BigData.2017.8258210.
9. D. Feng, C. Li, C. Xiao, and W. Sun, "Research of spectrum measurement of texture image," in World Automation Congress (WAC), 2012, pp. 163–165.
10. M. Padlia and J. Sharma, "Fractional sobel filter based brain tumor detection and segmentation using statistical features and SVM," in Lecture Notes in Electrical Engineering, 2019, vol. 511, pp. 161–175. doi: 10.1007/978-981-13-0776-8 15.
11. W. S. Rasband and others, "ImageJ," U.S. National Institutes of Health, Bethesda, Mary-land, USA, 1997. , imagej.nih.gov/ij/ (accessed Nov. 08, 2021).
12. K. Jafari-Khouzani and H. Soltanian-Zadeh, "Radon transform orientation estimation for rotation invariant texture analysis.," IEEE Trans Pattern Anal Mach Intell, vol. 27, no. 6, pp. 1004–8, Jun. 2005, doi: 10.1109/TPAMI.2005.126.
13. M. Kociolek, P. Bajcsy, M. Brady, and A. Cardone, "Interpolation-based gray-level co-occurrence matrix computation for texture directionality estimation," Signal Processing - Algorithms, Architectures, Arrangements, and Applications Conference Proceedings, SPA, vol. 2018-September, pp. 146–151, Dec. 2018, doi: 10.23919/SPA.2018.8563413.
14. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Commun ACM, vol. 60, no. 6, pp. 84–90, 2017.
15. S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," Proceedings - 30th IEEE Conference on Computer Vision and Pat-tern Recognition, CVPR 2017, vol. 2017-January, pp. 5987–5995, Nov. 2017, doi: 10.1109/CVPR.2017.634.
16. V. Fragoso, G. Srivastava, A. Nagar, Z. Li, K. Park, and M. Turk, "Cascade of box (CABOX) filters for optimal scale space approximation," IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp. 126–131, Sep. 2014, doi: 10.1109/CVPRW.2014.24.
17. R. Gonzalez and R. Woods, Digital Imaging Processing, 4th ed. Pearson, 2022. Accessed: Feb. 01, 2023. [Online]. Available: https://www.pearson.com/en-us/subject-catalog/p/digital-image-processing/P200000003224/9780137848560
18. M. Abadi et al., "TensorFlow: A System for Large-Scale Machine Learning," in 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), 2016, pp. 265–283. [Online]. Available: https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi
19. A. Gulli and S. Pal, Deep learning with Keras. Packt Publishing Ltd, 2017.
20. M. Kociolek and A. Cardone, marcinkociolek/DirDetectCORES2023, https://github.com/marcinkociolek/DirDetectCORES2023