Invariant Convolutional Networks

1st Matěj Lébl
Dept. of Image Processing
Czech Academy of Sciences,
Institute of Information Theory and Automation
Prague, Czech Republic
lebl@utia.cas.cz

2nd Jan Flusser
Dept. of Image Processing
Czech Academy of Sciences,
Institute of Information Theory and Automation
Prague, Czech Republic
flusser@utia.cas.cz

***Abstract*—Neural networks are often trained on datasets, that are not fully representative of the expected query images. Many times, the difference stem from the query images being taken in sub-optimal conditions. The most common defects are rotation, scale, blur, noise and intensity & contrast change which were all thoroughly studied and described. In this paper we propose a novel neural network architecture which is invariant to such degradations by design. We incorporate the knowledge build for classical methods directly into the network architecture providing an alternative to the augmentation of the training dataset. In the experiments, the proposed solution outperforms the classical augmentation technique in both accuracy and computational resources needed.**

I. Introduction

Over the past decade, the neural networks took over both industry and academia in terms of state-of-the-art problem-solving methods for many image processing tasks. Many of the models found their way into real world applications and lately, thanks to optimized architectures even into handheld devices. In many areas, neural network can even surpass human (e.g. plant classification based on leaf photo [2] ,[1]). However, like with any model, the quality of output is strongly dependent on the quality of the input image [3]. Images captured in unfavorable lighting conditions, with shaky camera or from different angle may fail to be classified (or segmented, tracked etc., in the rest of this paper we focus on classification). In case of a mobile app a simple warning text can prompt user to retake the photo (in a specific way) but in many cases this is not possible (e.g., images of stellar bodies, stained microscopic images etc.). Besides the engineering solution to this problem, the obvious step is to add various, on-purpose damaged images into the training set and re-run the training process. This paper proposes an alternative approach which rivals the training dataset augmentation approach without a need to alter or touch the existing, trained network. This allows for an attractive enhancement of preexisting systems and offers an interesting new angle for designing new architectures and further research.

II. Related work

Data quality and its impact on neural networks is well known and studied topic [5] ,[4] and more lately [7] ,[6]. Lebl et al. [3] showed, that even though effective, augmentation against blur must be carefully sampled over the expected extent of blur and over different blur types. The need for transformation-invariant features is extensively reviewed in [8].

The concept of invariants is quite old [9], see the book [10] for a survey and further references thereof.

Some recent works describe architectures, that are enforcing certain feature structure, which in return grants rotational invariance [12] [11]. In this paper we want to broaden the domain of invariant networks and extend them to arbitrary degradation, leveraging both handcrafted invariants and existing well-performing networks.

III. Method and Data

In this section we summarize a general theory of image invariants, establish (semantically) a new group of neural networks using those invariants and finally propose a novel alternative to dataset augmentation for dealing with degradations in query data.

For any computer vision task, real life application face challenges with data quality. Ranging from simple lack of sufficient training dataset for medical applications where one often must work with just couple dozens of labeled images to more traditional image degradations like rotation, scale, warping and blur - for example in by-leaf plant classification. Different strategies were developed over the years to counter these obstacles from engineering solutions (asking the user to control input image quality) to using separate CycleGANs to prepare arbitrary number of synthetic images covering majority of possible cases (e.g., stained tissue for cell classification and segmentation). The common theme is to either eliminate the problem or include samples of problematic data into the training set. Depending on the application, this can become quite time consuming if not near impossible given the extent of the augmentation needed (e.g., capturing sufficient range of blur families and levels).

A. Theory

An invariant I to a function f is a transformation with the property:

$$I(x) = I(f(x))$$

This work was supported by the Czech Science Foundation (Grant No. GA03921-21S) and by the *Praemium Academiae*.

979-8-3503-2541-6/23/$31.00 ©2023 IEEE

In the rest of the paper, we consider x being a clean, ideal image and f being some form of degradation - a transformation detrimental for a classification task. The entity $I(\cdot)$ may or may not in general be an image itself. We use the word invariant for both the operator and the resulting image. While invariants can be extremely powerful tool in image processing tasks, they tend to sacrifice the discriminability potential in favor of being agnostic to certain degradation. We are considering both of these properties of invariants and leveraging them as benefits.

Let's take the classification problem with a training/validation sets of ideal, clean images and a method (network) that solves the problem ideally - achieving maximal possible accuracy while having no redundancy in the model. Now we introduce a degradation to the input images which has detrimental effect on the model accuracy. Our proposed approach is to find and construct an invariant I and enhance the original method (without changing it) such that it solves the classification problem even with degraded input images.

B. Implementation

An obvious solution to the introduction of the degraded images to the classification problem is to simply include them also in the training set and retrain the model (augmentation). This serves as a baseline for comparison with our proposed solution. In addition, we set the following assumptions and constrains to create a level playing field:

- Base model has a satisfactory accuracy on the initial clean dataset. Complexity of the model is arbitrary (potentially very high) and training can use substantial resources.
- Degradations have detrimental impact on accuracy, they are broad in terms of range and complexity - making them hard to exhaustively sample, however they are well understood by classical methods.
- An invariant I to those degradation exist, describing the whole family of said degradations. It generally loses in discrimination power. This invariant should stay in the image domain (this condition can be relaxed) and should not exceed the original image in terms of complexity (and dimensions).
- We care about resources consumed, meaning we can't keep broadening the augmentation until we succeed or exhaust the potential of the given model.

We call the solution to the problem stated above the Invariant Network (INET). INets must satisfy the following conditions:

- Preserve accuracy on the original clean dataset.
- Greatly improve the accuracy on degraded data - on par with data augmentation.
- Agnostic to specific sub-type of given degradation.
- Use up at most the same computational resources as augmentation.

The third condition means for example invariance to ALL rotations, not only to rotation by 90, 180 and 270 $\deg$ or more practically to multiple blur families, not just Gaussian blur.

Four approaches were explored in total:

1) Building invariant representation of input data and feeding it directly to existing net.
2) Building network such that it is agnostic to specific type of degradation by design
3) Pre-processing the input while using the original architecture for classification
4) Using domain knowledge from handcrafted methods to support original architecture with an invariant extension

The first two options won't be discussed further in this paper. If invariant representation of an image preserves discriminability while being agnostic to certain degradation, there is no advantage of using the original image in the first place and any modification would increase resource consumption. Example being the invariant equals a bi-spectrum of an image and degradation equals rotation (there is no information loss in translating to bi-spectrum and CNN can be trained to work purely in frequency domain attaining the same accuracy albeit the computational cost would increase, similar idea explored by [13]). The second approach can be the best in certain situations and is explored for example by [11] in case of rotations, however the model needs to be constructed from scratch.

For this paper we've prepared two examples of invariant network architectures. The first proposal - SINet - maximizes the usage of the existing assets and tries to enhance the input to achieve invariance (approach 3.). We achieve this by simply adding another channel to the input (extreme case of image pre-processing) and re-use the original NN architecture unchanged. If the invariant representation is relatively close in image space to the original, we can expect reasonably good accuracy without any further action. By re-training the fully connected output layer, we aim to boost INET's ability to make a correct decision in case the confidence is low. By retraining the whole network, we can achieve even better results while still avoiding full augmentation of training dataset (as various damaged images are mapped to the same invariant).

The second proposal - πNet - is to prepare a separate, simpler network that is operating purely on the invariant representations of images (approach 4.). Features of this network are then combined with the features of the original, already trained network and final decision-making layer(s) is(are) added. This architecture originates from the idea of ensemble models with two main differences - it does not combine the final results but rather some low-level feature vectors and it only has one common loss function.

Note: It is obviously possible to prepare multiple different models and use sophisticated (e.g., machine learning) methods to combine results or build decision trees or even include the input image as a factor in choosing which solution to pick. This is commonly used in many areas: [14], [15]. However, we aim to provide simple, compact solution and stay strictly in the neural network domain.

C. Proposed solutions (INets)

Stacked invariant networks (SINet)

- Invariant must have the same dimensions as the original image
- Invariant must remain in the image domain
- Invariant added as a channel to input
- Re-training needed (reuses original weights for initialization)

Figure 1 shows the SINet architecture. The main strength of this approach is the simplicity and speed of implementation. We only care about constructing the invariant itself, the rest stays the same.

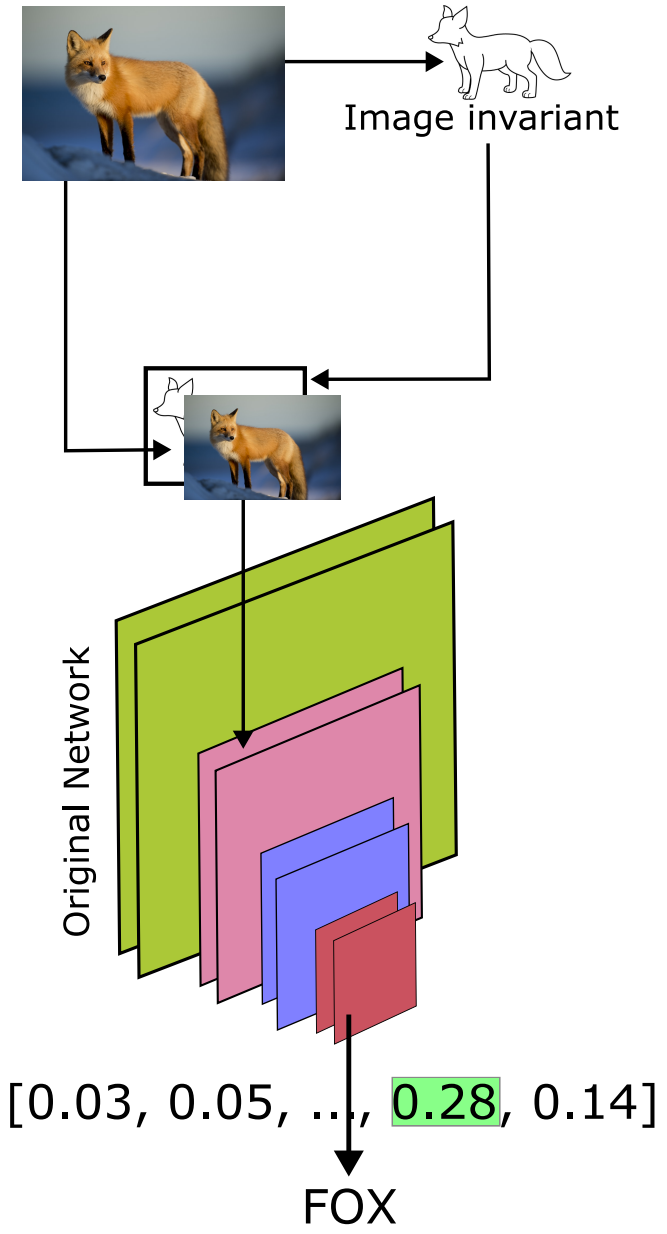

Fig. 1. SINet architecture - the original network is unchanged, input image is enriched by another channel - the invariant.

Parallel invariant network (πNet)

- Parallel network architecture with two branches merging into single output, see Fig.2
- Any invariant representation
- Invariant fed to separate branch
- Fully reuses original Net, up to final fully-connected layer
- Invariant branch can be pre-trained

As shown on Fig 2, the original neural network is unchanged up to the last fully connected layer (with all the weights frozen). The invariant branch can be pre-trained and have only the last layer fine-tuned to a specific task. Features of both branches are merged together, and another block is used before classification - this can be a single fully-connected layer or arbitrarily complicated sub-net.

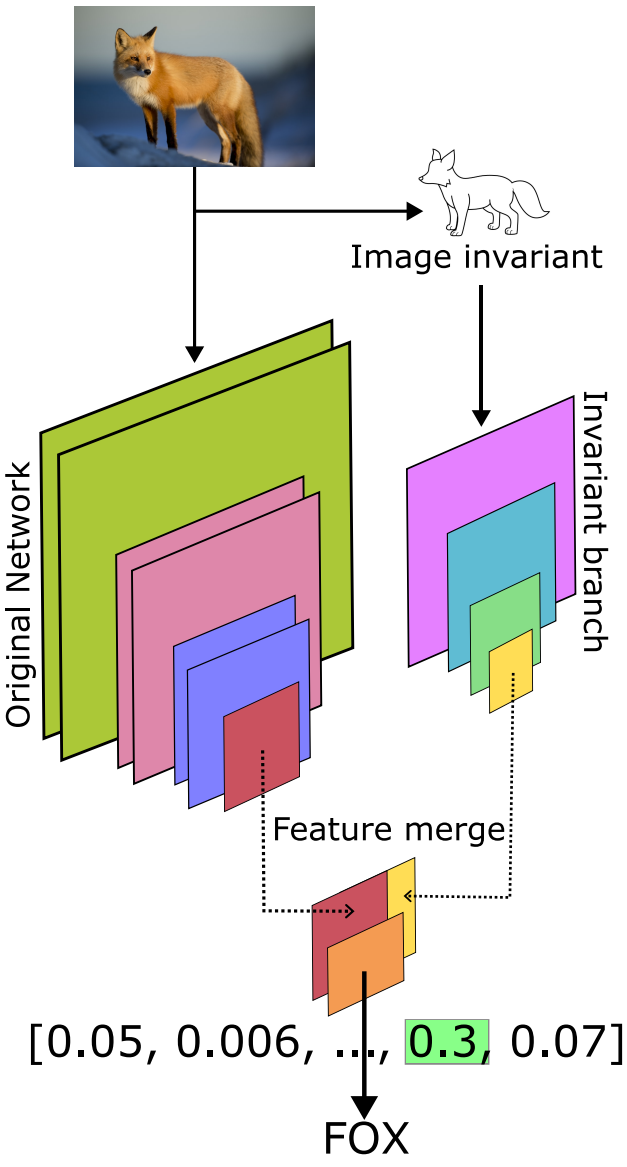

Fig. 2. πNet architecture - the original network (left branch) is unchanged up to the last layer(s), input image is converted into invariant and both branches - original and invariant (right) are merged before the final classification layer.

D. Summary

Existing original Network:

- Great accuracy on clean dataset
- Complex architecture
- Fine-tuned to specific problem
- Costly training

Degradations:

- detrimental impact on accuracy
- broad range
- well described by handcrafted methods
- difficult to parameterize (limited for rotations)
- difficult to exhaustively sample (limited for rotations)

Invariants:

- Describe whole family of degradations
- Stay in the image domain
- Substantial loss in discrimination power (limited for corruption of high frequency)

Invariant networks:

- Unaffected performance on clean dataset
- Greatly improved performance on degraded data
- Agnostic to degradation type (within a family)
- Competitive to augmentation

Baseline - full augmentation of training dataset:

- Sampling the space of degradations must be decided beforehand
- Achievable accuracy limited by architecture (number of parameters) in case of optimized networks
- Can be costly

IV. Experiments

We have designed two sets of experiments comparing accuracy of four models in seven different training configurations. We chose MNIST as a dataset for its simplicity and to have (nearly) perfect baseline accuracy even with simpler network architectures. The nature of the images also helps to select invariants designed as continuous functions even when dealing with discrete space, sampling, and finite precision. Training dataset contains 60000 images, validating 20000 images, image size is 28x28 px. Examples of images are in Fig. 3

Fig. 3. MNIST dataset samples - 4, 0, 8

First experiment uses randomly rotated data - each image from the validation dataset is rotated by a random angle (uniform distribution 0-359 $\deg$). Invariant is chosen as an average of all possible rotations, see fig 4.

There are many full rotational invariants - meaning the original image can be restored from the invariant (up to the rotation), one example being a bi-spectrum. We could use this invariant and achieve similar or better results than with proposed concentric circles, however full invariants are generally difficult (time consuming) to calculate, not robust in discrete space and finite precision and would require much more complicated invariant branch. Concentric circles are a straightforward method which is robust and compress n $\times$ n pixels to $\sim$ n/2 pixels. Such representation allows for simple architecture as we've reduced the complexity of input data.

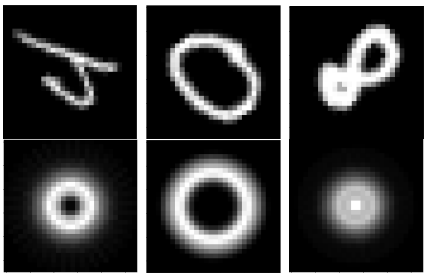

Fig. 4. MNIST dataset numbers 4, 0, 8. Top: rotated images, Bottom: rotational invariants - concentric circles

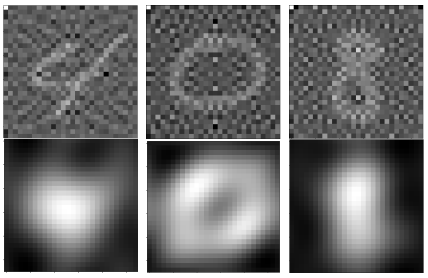

Fig. 5. MNIST dataset numbers 4, 0, 8. Top: noisy images - spectral damage, Bottom: spectral invariants - low-pass filter

Second uses corruption of high frequencies that looks like a noise in the image domain. First, we use Fourier transform, then all frequencies outside 2px radius have 25% probability to be damaged: equally likely either set value to 0 or set value to 1/4 of the zeroth frequency. Finally inverse Fourier transform brings the image back to image domain, but noisy. This way we can construct true invariant, which is a low-pass (2px radius) filtered image, see fig 5.

The models are:

- CNN - Original model solving given problem. Minimal convolutional neural network architecture such that attained accuracy is not improving anymore by adding more parameters (layers)
- SINet - the same network as above but takes two-channel input - image and its invariant representation, the initial weights are reused from the original CNN
- ICNN - control model - original CNN but the input is converted to the invariant representation
- πNet - Parallel network architecture with original branch being the original CNN and invariant branch having similar convolutional architecture (but fewer blocks). Outputs are concatenated and fed into fully connected layer.

Configurations are:

- Baseline - training done only on clean dataset.
- 100% augmentation - with and without including the full clean dataset, the full rotated / noisy dataset is used. Note that We re-trained all models twice, once for both kinds of degradations and corresponding invariants.
- 20% augmentation - only 20% of rotated / noisy datasets were used. With and without including the full clean dataset.
- 5% augmentation - same as above but only 5% of rotated / noisy datasets were used.

This setting always compares result achieved with similar level of computational effort invested and ensures as fair comparison as possible. See section IV-C for all the results side-by-side, we discuss the outcomes in more detail in subsections below.

A. Rotation invariance

Focusing on rotation-invariant, we achieve mediocre accuracy when using the pure invariants as an input (68%, see Tab I), this was an expected trade-off for an easy-to-calculate lightweight representation. However, by analyzing the Top3 accuracy of both the original network and the invariants we can build a traditional decision model that assigns scores to all classes based on joint confidences. Even without learning the weights, one can put together system that greatly outperforms the original network on rotated data and comes on par with it on clean data. Leaving the decision on the networks works for both INets and no further adjustments were necessary to rival augmentation as seen in table II. From the rest of the results, it is apparent that πNet is closely but consistently outperforming the other models in all configurations. This is more pronounced when we limit the resources - tables III and IV.

This setting fulfills all the requirements we set above and provides encouragement for further exploration of proposed approach. Mainly it proves that we don't need to solve the given degradation entirely and even simple naive invariants can greatly improve the overall accuracy.

B. Noise invariance

With the second degraded dataset and corresponding invariant, we deviated a little from the conditions set above to present more realistic problem that has actual detrimental effect on the recognition capability even for humans - heavy noise (realized as a damage in the spectral domain). The corresponding invariant was chosen to be a low-pass filter, with radius 2px.

Note: we tried different radii and we've obtained similar or better result even for low-pass radii up to 10px. We are presenting the results for the smallest - thus the most compact representation - radius that still yielded satisfactory results. It is possible to denoise the source images and omit the invariant architecture however that is not a topic of this paper.

To control this experiment and ensure both numerical stability and to guarantee invariance, we are damaging the spectrum center-symmetrically (so the inverse FFT is real-valued) and we don't damage the lowest frequencies (in the 2px radius). This can be related to older compression algorithms where images were sent first in low resolution and gradually downloaded more and more details - the initial low-res image is sent with high priority, ensuring save arrival while the rest is (more) likely to be corrupted. Limiting the low pass to those un-corrupted frequencies guarantees invariance. Compared to the rotation, heavily damaged images are hard to recognize even for humans while the invariants are somewhat readable, effectively being blurred versions of the original. This is reflected in the performance. The un-augmented original CNN is virtually useless because the input images are far from the training set while it reacts well to the invariants - using the pure invariants as an input yields 92% accuracy, see Tab I - as they resemble the original images. Mixing clear and damaged images during training then produces the desired result of almost unchanged accuracy on clear images and great accuracy on damaged images. Because of the nature of the damage, the original CNN was not able to fully learn the degraded data, even with heavy augmentation while πNet attained excellent results even with only partial (20%) augmentation.

Note: there is a drop in accuracy when training only on degraded images at 45% of the training dataset where the original CNN is unable to train and stays at the initial 11% accuracy.

C. Result tables

Baseline: Both proposed solutions maintain great accuracy on the clean dataset however without degraded data in the training set, the invariant branch is not given significant weight. Noise invariants alone perform well on degraded dataset but fall short on clean images.

	Rotation		Noisy	
Architecture	clean	damaged	clean	damaged
CNN	99%	44%	99%	11%
SINet	**99%**	44%	**99%**	11%
ICNN	68	**68%**	92%	**92%**
πNet	99%	45%	99%	39%

TABLE I
ALL 4 MODELS TRAINED ON 100% OF THE CLEAN DATASET AND VALIDATED ON CLEAN, ROTATED AND NOISY DATASET.

Trained on clean + degraded dataset (2x 100% of training data): Both proposed solutions achieved our goal - almost unaltered performance on clean dataset and satisfactory accuracy on degraded images. Note that pure training dataset augmentation was not enough to reach good accuracy for CNN model.

	Rotation		Noisy	
Architecture	clean	damaged	clean	damaged
CNN	98%	93%	98%	70%
SINet	98%	93%	98%	92%
ICNN	69%	69%	93%	93%
πNet	**98%**	**94%**	**98%**	**94%**

TABLE II
ALL 4 MODELS TRAINED ON 100% OF THE CLEAN + 100% OF DEGRADED DATASET AND VALIDATED ON CLEAN, ROTATED AND NOISY DATASET.

Trained on clean + degraded dataset (100% + 20% of training data): Performance is slightly better on clean images than in table II but we lose accuracy on degraded images.

	Rotation		Noisy	
Architecture	clean	damaged	clean	damaged
CNN	**99%**	90%	**99%**	66%
SINet	99%	90%	99%	89%
ICNN	67%	67%	92%	92%
πNet	99%	**92%**	99%	**93%**

TABLE III
ALL 4 MODELS TRAINED ON 100% OF THE CLEAN AND 20% OF DEGRADED DATASET AND VALIDATED ON CLEAN, ROTATED AND NOISY DATASET.

Trained on clean + degraded dataset (100% + 5% of training data): All models start rapidly losing accuracy on degraded images compared to Table II.

	Rotation		Noisy	
Architecture	clean	damaged	clean	damaged
CNN	99%	80%	99%	42%
SINet	**99%**	80%	**99%**	87%
ICNN	67%	67%	92%	92%
πNet	99%	**85%**	99%	**92%**

TABLE IV
ALL 4 MODELS TRAINED ON 100% OF THE CLEAN AND 5% OF DEGRADED DATASET AND VALIDATED ON CLEAN, ROTATED AND NOISY DATASET.

Inspecting the results above once can deduce that we are also able (to some extent) control the bias towards either degraded images or clean images. With full training dataset augmentation we sacrifice a little bit of accuracy on clean images and gain great accuracy on degraded data. However, if we expect majority of our data to be clean, we recommend augmenting only portion of the training set.

V. CONCLUSION

The above experiments justify the proposed solution and encourage further exploration of πNet architecture.

The biggest benefits of πNet are: Great accuracy, easy integration into existing applications and the training cost ceiling - for each training image there is exactly one invariant, making the training dataset at most twice as big. Comparing to classical augmentation, where we may need to add multiple different instances of degradations for each training image. The original, already trained network can be fully utilized up to the fully connected output layer. The invariant branch of πNet can be pre-trained just like any other network using for example ImageNet so we only need to re-train two FC layers and train the final output block. We thus have an upper bound on the training complexity needed and as shown in the experiments, we can even reduce the augmentation rate in case the resources are scarce. This is especially useful in commercial applications, where knowing the expected load beforehand is crucial for estimating a budget.

We have proposed a novel approach for handling degradations of input data for neural networks which rivals and even surpasses augmentation with similar or smaller computational resource investments.

With the proposed πNet we achieved:

- Comparable performance as base CNN on clean data
- Best performance on degraded data out of all tested architectures
- Agnostic to base CNN - plug-and-play architecture
- Competitive cost of training

As a next step we plan to generalize the framework further as well as suggest optimized strategies for designing the invariant branch. Second area is to combine multiple invariants in a single step and prepare architecture robust to two or more degradations at the same time.

REFERENCES

[1] Guillermo L Grinblat, Lucas C Uzal, Mónica G Larese, and Pablo M Granitto, "Deep learning for plant identification using vein morphological patterns," *Computers and electronics in agriculture*, vol. 127, pp. 418–424, 2016.

[2] Boi M Quach, V Cuong Dinh, Nhung Pham, Dang Huynh, and Binh T Nguyen, "Leaf recognition using convolutional neural networks based features," *Multimedia Tools and Applications*, vol. 82, no. 1, pp. 777–801, 2023.

[3] Matěj Lébl, Filip Šroubek, and Jan Flusser, "Impact of image blur on classification and augmentation of deep convolutional networks," in *Image Analysis: 23rd Scandinavian Conference, SCIA 2023, Sirkka, Finland, April 18–21, 2023, Proceedings, Part II*. Springer, 2023, pp. 108–117.

[4] Luis Perez and Jason Wang, "The effectiveness of data augmentation in image classification using deep learning," *arXiv preprint arXiv:1712.04621*, 2017.

[5] Maayan Frid-Adar, Idit Diamant, Eyal Klang, Michal Amitai, Jacob Goldberger, and Hayit Greenspan, "Gan-based synthetic medical image augmentation for increased cnn performance in liver lesion classification," *Neurocomputing*, vol. 321, pp. 321–331, 2018.

[6] Luke Taylor and Geoff Nitschke, "Improving deep learning with generic data augmentation," in *2018 IEEE symposium series on computational intelligence (SSCI)*. IEEE, 2018, pp. 1542–1547.

[7] Qinghe Zheng, Mingqiang Yang, Xinyu Tian, Nan Jiang, Deqiang Wang, et al., "A full stage data augmentation method in deep convolutional neural network for natural image classification," *Discrete Dynamics in Nature and Society*, vol. 2020, 2020.

[8] Alhassan Mumuni and Fuseini Mumuni, "Cnn architectures for geometric transformation-invariant feature representation in computer vision: a review," *SN Computer Science*, vol. 2, pp. 1–23, 2021.

[9] David Hilbert, *Theory of Algebraic Invariants*, Cambridge University Press, Cambridge, U.K., 1993.

[10] J. Flusser, T. Suk, and B. Zitová, *2D and 3D Image Analysis by Moments*, Wiley, Chichester, U.K., 2016.

[11] Daniel E Worrall, Stephan J Garbin, Daniyar Turmukhambetov, and Gabriel J Brostow, "Harmonic networks: Deep translation and rotation equivariance," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5028–5037.

[12] Benjamin Chidester, Tianming Zhou, Minh N Do, and Jian Ma, "Rotation equivariant and invariant neural networks for microscopy image analysis," *Bioinformatics*, vol. 35, no. 14, pp. i530–i537, 2019.

[13] Jinpyo Kim, Wooekun Jung, Hyungmo Kim, and Jaejin Lee, "Cycnn: A rotation invariant cnn using polar mapping and cylindrical convolution layers," *arXiv preprint arXiv:2007.10588*, 2020.

[14] Danish Vasan, Mamoun Alazab, Sobia Wassan, Babak Safaei, and Qin Zheng, "Image-based malware classification using ensemble of cnn architectures (imcec)," *Computers & Security*, vol. 92, pp. 101748, 2020.

[15] Emanuela Paladini, Edoardo Vantaggiato, Fares Bougourzi, Cosimo Distante, Abdenour Hadid, and Abdelmalik Taleb-Ahmed, "Two ensemble-cnn approaches for colorectal cancer tissue type classification," *Journal of Imaging*, vol. 7, no. 3, pp. 51, 2021.