Classification of UAVs' distorted images using Convolutional Neural Networks

Leandro H. F. P. Silva*, Jocival D. D. Júnior*, Jean Fabrico Batista Santos*, João F. Mari†,
Mauricio C. Escarpinati*, André R. Backes*,
*School of Computer Science, Federal University of Uberlândia, Brazil
†Federal University of Viçosa, Brazil
arbackes@yahoo.com.br

***Abstract*—Currently, the use of unmanned aerial vehicles (UAVs) is becoming ever more common for acquiring images in precision agriculture, either to identify characteristics of interest or to estimate plantations. However, despite this growth, their processing usually requires specialized techniques and software. During flight, UAVs may undergo some variations, such as wind interference and small altitude variations, which directly influence the captured images. In order to address this problem, we proposed a Convolutional Neural Network (CNN) architecture for the classification of three linear distortions common in UAV flight: rotation, translation and perspective transformations. To train and test our CNN, we used two mosaics that were divided into smaller individual images and then artificially distorted. Results demonstrate the potential of CNNs for solving possible distortions caused in the images during UAV flight. Therefore this becomes a promising area of exploration.**

***Index Terms*—Convolutional Neural Networks, Precision Agriculture, Unmanned Aerial Vehicle, Linear Distortions, Image Processing**

I. Introduction

At the end of the 19th century, studies already indicated concern about the growth of the world population and the capacity of the planet to produce food to feed it [1]. At the time it was feared that the population would grow in geometric progression, while food production would grow in arithmetic progression. In the end, this would lead to a drastic food shortage and, as a consequence, hunger. Therefore, inevitably population growth should be controlled.

These predictions were not confirmed, largely due to the significant technological advances that occurred in the agricultural area between 1950 and the late 1960s [2], a set of research technology transfer initiatives that increased agricultural production worldwide, particularly in the developing world, beginning most markedly in the late 1960s. The initiatives resulted in the adoption of new technologies, including high-yielding varieties (HYVs) of cereals, especially dwarf wheats and rices, in association with chemical fertilizers and agro-chemicals, and with controlled water-supply (usually involving irrigation) and new methods of cultivation, including mechanization. All of these aspects mentioned were seen as a kind of "practice package" to replace "traditional" technology and thus being used as a whole [3].

Nowadays, we are seeing a new evolutionary phase in the field of agriculture researches. The main component of this phase is Precision Agriculture (PA), which is nothing more than a farming management concept based on observing, measuring and responding to inter and intra-field variability in crops. The goal of precision agriculture research is to define a decision support system (DSS) for whole-farm management with the goal of optimizing returns on inputs while preserving resources [4].

Dealing specifically with problems involving the PA area, it has shown itself to be heavily dependent on imaging and mapping technologies e.g. for estimating growth [5], or identifying other important agronomic characteristics [6] such as nitrogen stress. Advances in Unmanned Aerial Vehicles - UAV - technology led to its widespread popularization. With the corresponding drop in operational costs even smaller plantations are now able to afford the usage of imaging aided technologies. The latest economic report by the Association of Unmanned Aerial Vehicle International [7] points out the agricultural market is by far the largest segment for UAVs. In the United States alone is forecast to create thousands of new jobs and considerable revenue and taxes. With the growth of this market production costs are expected to drop. It, in turn, will allow smaller enterprises such as family and small agricultural cooperatives [8] to benefit from the diminished operational costs to also make use of precision agriculture aided by UAVs. Other countries like Japan are also making extensive use of UAVs in agriculture and in Brazil there is a growing number of startup companies producing and commercializing UAVs.

Different from all other aerial image acquisition devices, such as satellites and large aircraft, UAV's allow images to be captured at low and medium altitudes (50 to 400 m), providing a more detailed view of the region to be observed. Another important element for the effectiveness of the analysis performed with this equipment is the used sensors. There is a wide range of devices used in the process: RGB cameras; heat capture sensors, multi and hyperspectral cameras, among others. Each device, with its characteristics, produces information that leads to different types of analysis. However, the process of data acquisition, in general, is the same independent of the sensor used: the equipment is coupled to the aircraft and the images are sequentially captured during the flight. After finishing the process, with the aircraft already on the ground, these images are organized into a mosaic to represent the entire area.

As the images are taken during the flight, UAVs may

undergo some variations, such as wind interference and small altitude variations, which directly influence the captured images, causing a natural misalignment among the imagens that comprise the mosaic and, more often, among the different spectra witch form a specific frame. Usually, the distortion generated in this process are classified as linear distortion and can affect significantly the success of specific software used in agriculture images. Thus, in order to address this problem, the present work proposes a Convolutional Neural Network (CNN) trained for the classification of three linear distortions common in UAV flight: rotation, translation and perspective transformations.

The remainder of this paper is organized as follows. Section II shows some recent papers published in the area. In Section III we detail the problem and their implications. In Section IV, we present an overview of the CNN and how it was used to deal with our problem. Section V presents the image dataset used in the experiments. Sections VI and VII present the experiments and a discussion of the results. Section VIII presents the conclusions and future work.

II. RELATED WORK

In [9] the authors evaluated different techniques for obtaining control points in multispectral images of soy plantations obtained by UAVs. The authors also investigated whether the combination of characteristics derived from different techniques generates better results than when those techniques are used individually. The paper evaluated three detection algorithms with different characteristics (KAZE, MEF, and BRISK) and their combinations. Results show that KAZE techniques have the best results.

In [10] the authors presented a convolutional neural network to estimate homography from a pair of images. The network in question has 10 layers with feed-forward architecture and receives a pair of grayscale images. Subsequently, it produces a homography with 8 degrees of freedom, which can be used to map the pixels from the first to the second image.

The work in [11] introduces a hierarchical approach based on Siamese convolutional neural networks to estimate homography between two images. The networks are stacked sequentially to estimate of error limits. In each convolutional network module, the resources of each image are extracted independently, generating a shared set of kernels, which is known as the Siamese model. Subsequently, the image pairs are merged to estimate the homography. With this approach, the results show that through deep learning it is possible to estimate homography from an image pair.

III. PROBLEM DEFINITION

Due to the inherent aspects of UAV flight, image capture is subject to distortion that needs to be dealt with and corrected. These distortions may be linear or nonlinear. In this paper, we will consider only three linear distortions that may occur during flight: translation, rotation and perspective transformation.

In a translation operation all points are moved in a straight line in the same direction. In summary, a conversion operator will perform a geometric transformation that maps the position of each element of the image in an input image to a new position in the output image [12].

Rotation transformation is defined as a rotary movement on a fixed axis. According to Gonzalez (2002) [12], three transformations are needed to rotate a point relative to another arbitrary point in space: the first will translate the arbitrary point to the origin, the second will rotate, and finally the third will translate the point back to its original position.

A perspective transformation in general takes place with the conversion of the 3D world into a 2D image. This is the same principle that human vision works on and the same principle that the camera works on. In perspective projections, parallel lines converge (in 1, 2, or 3 axes) for a given point. This way, objects that are farther away are smaller than closer objects.

Perspective transformation will project three-dimensional points onto a plane. Such transformations play a fundamental role in image processing, as they offer a way of approximating the way in which the image is formed by looking at the three-dimensional world [12]. In general, these projective transformations allow us to capture natural motion dynamics through a mathematical mechanism. These transformations do not preserve size or angle but preserve incidence and cross-ratio.

IV. CONVOLUTIONAL NEURAL NETWORK

Convolutional Neural Networks (CNN) are a category of deep learning algorithms capable to mimic the human learning process. These networks are based on the concept of the receptive field from biological systems, which gives these networks the ability to learn different filters and characteristics from an image. This way, CNN can explore the spatial correlations among pixels in an image in order to extract image attributes that are relevant for different tasks, such as image classification and segmentation [13]–[15]. Most CNN models available in the literature are defined in terms of three types of layers, which are differently combined to improve image classification or segmentation: convolutional, pooling and fully connected layer. In the sequence, we present a brief description of each layer.

The convolutional layer is responsible for extracting meaningful attributes from an image. To accomplish that, it applies a series of convolution operations to the input data, which acts as receptive filters that highlight different attributes of a local region of the image. In general, theses filters are defined as kernels size 3×3 or 5×5. Additionally, the activation function ReLU (REctified Linear Unit) and a Batch Normalization operation are applied to the result of the convolutional layer. This helps to speed up the training of the network and to improve its results [16].

The convolutional layer is usually followed by a pooling layer. The main purpose of this layer is to reduce the feature maps computed by the previous layers, thus reducing the network sensitivity to distortions in the image and data shifting. In general, it is used a pooling mask of size 2×2, thus reducing

a 4 pixels region to a single value according to some criteria (e.g., maximum or the average pixel of the region) [17].

At the end of the CNN, we find the fully connected (or dense) layer. About 90% of the parameters of a CNN are found in these layers. This layer receives as input data the 2D features maps obtained from previous layers and its main goal is to learn a 1D feature vector capable to discriminate the input image. This feature vector is used as the input of a softmax classifier, which returns the most probable class for a given input image.

V. IMAGE DATASET

A. Selected Images

For our experiments, we considered two mosaics of images acquired using an unmanned aerial vehicle (UAVs) to create the datasets of images used in the experiments. These mosaics have 18543×2635 and 8449×11180 pixels size, respectively.

For each dataset, we selected grayscale patches of 150×150 pixels size. Subsequently, we discard patches that have little (or any) significant visual information. This was determined by the number of pixels (n) with a value of 0 in the patch. Thus, if $n < 20$, the patch is considered for the composition of the dataset; otherwise, the patch is discarded. Therefore, we built two datasets, which we will call DS1 and DS2 and which have, respectively, 3218 and 1586 images. Figure 1 illustrates two examples of images patches generated for each dataset.

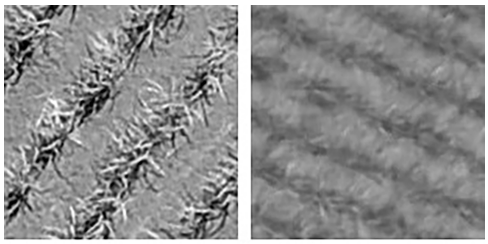

Fig. 1. Example of images that make up both datasets: (a) an image of DS1; (b) an image of DS2.

B. Dataset images distortions

For both datasets, DS1 and DS2, we artificially distorted the images using two affine (rotation and translation) and one projective (perspective) transformation. It is important to mention that, as a result of the transformation method, the transformed images have black areas, especially at the limits of the image area, which can directly influence the neural network training and testing (see in Figure 2). To avoid these black areas, we cropped a 64×64 pixels region aligned with the center of the image, thus removing any artifact added to the image by the selected transformation method.

In order to apply the transformation over the images, the following set of parameters were used:

- **Rotation:** we used $\theta = \{0°, 5°, 10°, 15°\}$, thus generating 4 classes of rotated patterns.

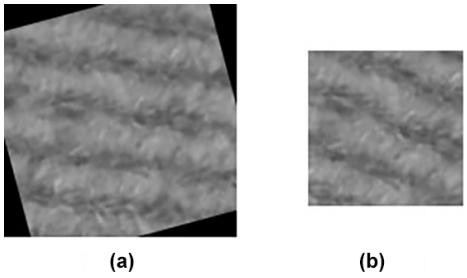

Fig. 2. (a) Image after a 15-degree rotation transformation. Notice that this image presents black areas which can directly influence the neural network training and testing; (b) Cropped region with 64×64 pixels size.

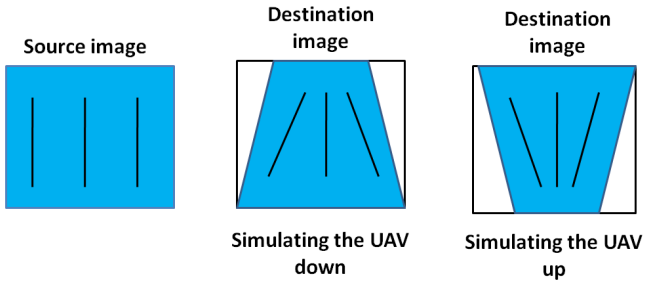

Fig. 3. Perspective transformation in UAV up and down simulations.

- **Translation:** images were translated by 25 pixels in 4 possible directions: right and top; right and down; left and top; and left and down, thus generating 5 equivalence classes (the original image is also included).
- **Perspective:** To simulate UAV up and down possibilities in moments of image capture, we also deal with perspective transformation. The Figure 3 illustrates the UAV up and down simulations and the respective distortions caused by pitch variations. For this transformation, we generated two variations for each of the two possibilities mentioned above, thus totaling 5 equivalence classes (the original image is also included). In this way, we choose four control points in a source image to map it to a destination image. Perspective transformation works with the row and column relationship. As we are only simulating the UAV up and down possibilities, we keep the proportion of lines identical to the original image. For the columns, the proportions in each of the distorted classes created were: (0.05, 0.66); (0.05, 0.77); (0.02, 0.66); (0.02, 0.77).

It is also necessary to define a mathematical operation that relates the distorted image to the base image, otherwise it is impossible to state that an image is distorted. Thus, all artificially distorted images underwent a subtraction operation from the original image. Let A be the distortion-free image and B the distorted image relative to A, we define X as the image resulting from the subtraction operation and to be processed by the CNN. The operation performed between A

and B is defined pixel by pixel. We must also consider that the subtraction operation may result in negative values and an image is expected to have only positive values. To avoid that, we normalized the computed x_{ij} values as follows:

$$x_{ij} = max(b_{ij} - a_{ij}, 0) \quad (1)$$

where $a_{ij} \in A$ represents a pixel of image A, $b_{ij} \in B$ represents a pixel of image B and $x_{ij} \in X$ represents a pixel of image X. Figure 4 illustrates a subtraction between an artificially distorted image (rotation) and a distortion-free image.

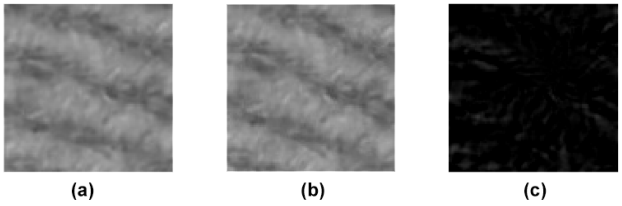

Fig. 4. Example of subtraction operation between two images: (a) Artificially distorted image (rotation); (b) Corresponding distortion-free image; (c) Result from the subtraction operation ($(c) = (a) - (b)$).

VI. EXPERIMENTS

We also carried out a data augmentation to reduce the possibility of overfitting in our experiments. In addition to the traditional CNNs, we proposed an alternative architecture that will be presented as follows. Our architecture is motivated by [18], [19], where simpler CNNs and sets of filters were used to solve less complex classification problems.

In order to address our image analysis problem, we proposed a network structure. Due to the reduced size of our samples (64×64 pixels size), our CNN presents fewer layers than conventional CNNs. To properly process our images we used a CNN with 5 convolutional layers. Each convolutional layer presents, respectively, 32, 64, 64, 128 and 256 filters. To improve the network performance and to speed up its training, we apply non-linearity ReLU activation function after each convolutional layer. We also apply a batch normalization after the ReLU filter, which is followed by a 2×2 max-pooling layer.

After the convolutional layers, we use the resulting volume ($2 \times 2 \times 256$ output shape and 1024 features) as input for the dense layers. The first and second dense layers have 128 neurons and the activation function ReLU. After each dense layer we applied dropout of 20%. Finally, the output layer has 4 or 5 neurons (4 for rotation; otherwise, 5 neurons) that determined the class, as we expound in subsection V-B.

To implement the convolutional neural networks used in this work we used the Python version of Tensorflow, an open-source library developed by Google [20] for efficient building, training and use of deep neural network models. TensorFlow is based on tensors and dataflow graphs. Tensors are numerical multidimensional arrays that represent the data. Dataflow graphs nodes represent operations while edges describe the flow of data throughout the processing steps. TensorFlow dataflow graphs are very modular and allow building complex models directly. These models can be trained and run in a myriad of environments taking advantage of the high parallelism of modern GPUs [21]–[23].

We evaluated our CNN model using both datasets, as defined in Section V. For each dataset we selected 75% of the samples to compose the training set, while the remaning images were used for validation. Motivated by work [24], we chose not to perform cross-validation for this purpose. The work [24] demonstrates that in problems in this context, the use of cross-validation does not generate much difference in the final results, except that it increases the computational cost considerably. Both datasets will be available for replication and other experiments as request.

Experiments were conducted on a Personal Computer with Intel(R) Core(TM) i7-7700 CPU @ 3.60GHz, 32GB RAM, 64-bit Windows OS and GPU NVIDIA GeForce GTX 1050 Ti, 4GB GDDR5. We also used Python 3.6 and Keras 2.1.6-tf with TensorFlow 1.10.0 and CUDA Toolkit 9.0 to implement and test the experiments.

VII. RESULTS

First, for each dataset (DS1 and DS2) we generated a new dataset with one of the specified distortions. Then, this new dataset was split between training and validation samples and used to train our CNN model for 20 epochs. After this, we are able to analyze the accuracy of our model for detecting the distortions analyzed.

We notice that the best performance is obtained when dealing with the problem of image rotation, as illustrated in Figure 5(a). For the rotation problem, our CNN model is capable to classify the rotation distortion with 99.85% and 99.18% accuracy in the DS1 and DS2 datasets, respectively. Moreover, the CNN presents a good ability to generalize the features learned in the training set to the test set. This may be explained by the fact that the rotation operation results in a less distorted image in comparison to other image transformations, i.e., an easier classification problem.

Figure 5(c) shows the performance for perspective transformation. For this transformation, our CNN model achieves high accuracy, especially for DS1, which reached 95.50%. For the same problem, however, we obtained only 91.47% for DS2 accuracy. Notice that this accuracy is substantially lower when compared to the rotation experiment. One explanation for this behavior is that this kind of transformation affects differently the regions of the sample, while rotation affects all points of the sample equally. Moreover, although both datasets present a lower result, we notice that dataset DS1 presents a superior result when compared to dataset DS2. It may be the case that the number of samples in the training set, which is larger in DS1, contributes positively to learn this transformation.

For the translation transformation, we noticed a considerable drop in the accuracy of the network when evaluating the DS2 dataset (70.24%), as shown in Figure 5(b). Even though the dataset DS1 (Figure 5(b)) also present an inferior performance

when compared to the rotation transformation, its result is superior to the ones obtained for the perspective transformation. This result observed in the DS2 dataset is probably explained by the lack of details in their original images, as illustrated in Figure 1. Since crop lines and land regions present similar gray-level distributions, the result of the subtraction operation between the original and the translated image results in a mostly black image, i.e., an image without enough attributes for our CNN to learn.

In order to improve the evaluation of our CNN model we compared its results with the ones obtained by 4 traditional CNN models: InceptionV3 [25], ResNet [26], SqueezeNet [27] and VGG-16 [28]. For this comparison we used pre-trained networks on the 2012 ImageNet dataset and fine-tuned the whole CNN to our classification problem for 20 epochs. We must emphasize that these networks have a input size larger than the samples in our datasets so that all images have been scaled up to fit the input size of the respective network.

Table I summarizes the results of all CNN models. As we can see, our CNN surpasses the results of all compared ones, indicating that its architecture, although simpler than the compared ones (see Table II), is more effective to classify images obtained from the difference of intensities between two images and, therefore, presenting a small variation of gray levels.

TABLE I
ACCURACY (%) OBTAINED FOR OUR CNN AND THE COMPARED ONES.

	Translation		Rotation		Perspective	
CNN model	DS1	DS2	DS1	DS2	DS1	DS2
ResNet	91.83	48.13	95.00	96.84	59.55	62.63
InceptionV3	20.00	60.10	98.48	98.23	20.00	65.96
VGG-16	94.76	65.15	98.63	98.74	84.89	75.20
SqueezeNet	90.51	40.40	91.77	96.15	55.68	55.20
Proposed	96.92	70.24	99.85	99.18	95.50	91.47

TABLE II
NUMBER OF PARAMETERS OF EACH CNN MODEL.

CNN model	# of parameters
ResNet	23,595,908
InceptionV3	22,082,084
VGG-16	14,797,380
SqueezeNet	725,061
Proposed	477,573

VIII. CONCLUSION

In this paper, we addressed the problem of classifying different types of distortions in images acquired using unmanned aerial vehicles (UAVs). To accomplish that we proposed and trained a Convolutional Neural Network (CNN) model to learn the subtleties that distinguish each transformation studied: translation, rotation and perspective transformation.

Results showed that our CNN model is capable to correctly classify the different transformations, especially the rotation transformation. However, the performance of the CNN is dependent on the image resolution and gray-levels distributions present in the sample image evaluated so that datasets containing blurry images affects negatively the performance of our network. Also, our architecture, due to its low computational cost, can inspire embedded systems to UAVs in the context of precision agriculture, reducing financial costs inherent to the process. As future work, we intend to expand the dataset used in the experiments and to include images containing real distortions produced during a UAV flight and to explore other models of CNN.

ACKNOWLEDGMENT

André R. Backes gratefully acknowledges the financial support of CNPq (National Council for Scientific and Technological Development, Brazil) (Grant #301715/2018-1). This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brazil (CAPES) - Finance Code 001. The authors would like to thank the company Sensix Inovações em Drones Ltda (http://sensix.com.br) for providing the images used in the tests.

REFERENCES

[1] T. R. Malthus, *An Essay on the Principle of Population..*, 1872.
[2] P. B. Hazell, *The Asian green revolution*. Intl Food Policy Res Inst, 2009, vol. 911.
[3] B. Farmer, "Perspectives on the 'green revolution'in south asia," *Modern Asian Studies*, vol. 20, no. 1, pp. 175–199, 1986.
[4] A. Milella, G. Reina, and M. Nielsen, "A multi-sensor robotic platform for ground mapping and estimation beyond the visible spectrum," *Precision agriculture*, vol. 20, no. 2, pp. 423–444, 2019.
[5] T. Kataoka, T. Kaneko, H. Okamoto, and S. Hata, "Crop growth estimation system using machine vision," in *Proceedings 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2003)*, vol. 2. IEEE, 2003, pp. b1079–b1083.
[6] S. Sankaran, L. R. Khot, C. Z. Espinoza, S. Jarolmasjed, V. R. Sathuvalli, G. J. Vandemark, P. N. Miklas, A. H. Carter, M. O. Pumphrey, N. R. Knowles *et al.*, "Low-altitude, high-resolution aerial imaging systems for row and field crop phenotyping: A review," *European Journal of Agronomy*, vol. 70, pp. 112–123, 2015.
[7] D. Jenkins and B. Vasigh, *The economic impact of unmanned aircraft systems integration in the United States*. Association for Unmanned Vehicle Systems International (AUVSI), 2013.
[8] J. M. Turner, "Economic potential of unmanned aircraft in agricultural and rural electric cooperatives," Ph.D. dissertation, 2016.
[9] J. D. D. Junior, A. R. Backes, and M. C. Escarpinati, "Detection of control points for uav-multispectral sensed data registration through the combining of feature descriptors," 2019.
[10] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Deep image homography estimation," *arXiv preprint arXiv:1606.03798*, 2016.
[11] F. Erlik Nowruzi, R. Laganiere, and N. Japkowicz, "Homography estimation from image pairs with hierarchical convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 913–920.
[12] R. C. Gonzalez, R. E. Woods *et al.*, "Digital image processing," 2002.
[13] Y. L. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[14] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, "Deep learning for visual understanding: A review," *Neurocomputing*, vol. 187, pp. 27–48, 2016.
[15] M. A. Ponti, L. S. F. Ribeiro, T. S. Nazaré, T. Bui, and J. Collomosse, "Everything you wanted to know about deep learning for computer vision but were afraid to ask," in *SIBGRAPI Tutorials*. IEEE Computer Society, 2017, pp. 17–41.
[16] Y. LeCun, Y. Bengio, and G. E. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

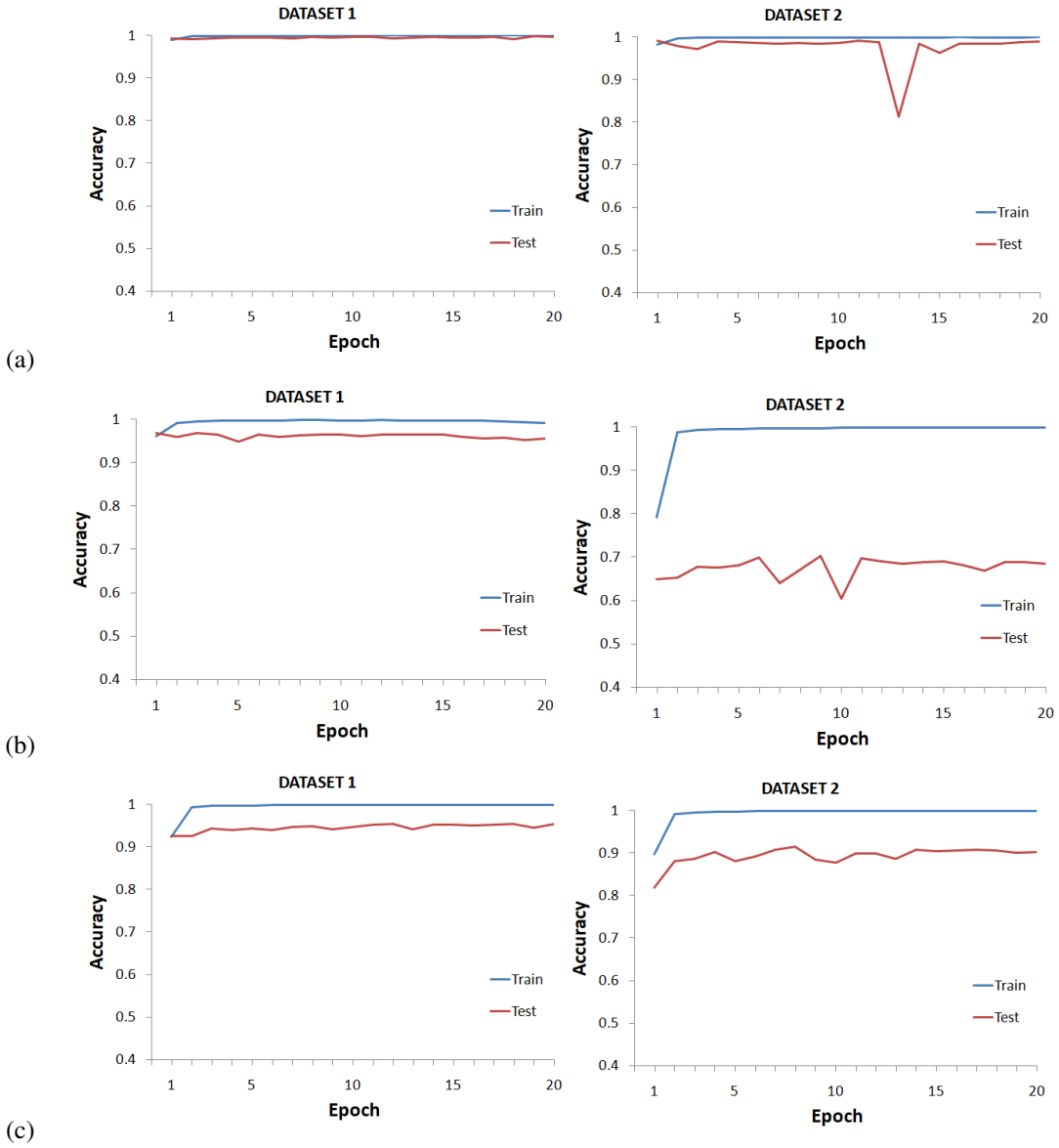

Fig. 5. Accuracy of our CNN to classify distortions in both datasets: (a) Rotation; (b) Translation; (c) Perspective.

[17] D. Scherer, A. C. Müller, and S. Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition," in *Artificial Neural Networks - ICANN 2010 - 20th International Conference, Thessaloniki, Greece, September 15-18, 2010, Proceedings, Part III*, ser. Lecture Notes in Computer Science, vol. 6354. Springer, 2010, pp. 92–101.

[18] A. P. Marcos, N. L. S. Rodovalho, and A. R. Backes, "Coffee leaf rust detection using genetic algorithm," in *2019 XV Workshop de Visão Computacional (WVC)*. IEEE, 2019, pp. 16–20.

[19] ——, "Coffee leaf rust detection using convolutional neural network," in *2019 XV Workshop de Visão Computacional (WVC)*. IEEE, 2019, pp. 38–42.

[20] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: A system for large-scale machine learning," 2016.

[21] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, 2019.

[22] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[23] T. Hope, Y. S. Resheff, and I. Lieder, *Learning tensorflow: A guide to building deep learning systems*. " O'Reilly Media, Inc.", 2017.

[24] A. R. de Geus, A. R. Backes, and J. R. Souza, "Variability evaluation of cnns using cross-validation on viruses images." in *VISIGRAPP (4: VISAPP)*, 2020, pp. 626–632.

[25] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[27] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <0.5mb model size," *arXiv:1602.07360*, 2016.

[28] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.