

J. Comp. Int. Sci. (2019) 10(3):145-154  
[http://epacis.net/jcis/PDF\\_JCIS/JCIS-0168.pdf](http://epacis.net/jcis/PDF_JCIS/JCIS-0168.pdf)  
[jcis@epacis.net](mailto:jcis@epacis.net)  
©2019 PACIS (<http://epacis.net>)



PAN-AMERICAN  
ASSOCIACION OF  
COMPUTATIONAL  
INTERDISCIPLINARY  
SCIENCES

## Evaluating the Deep Learning accuracy in data extraction from synthetic image sequences

André Franceschi de Angelis<sup>1</sup> and Thaís Rocha

University of Campinas, School of Technology, Brazil

Received on Dec 15th, 2018 / published on Nov 30th, 2019

---

### Abstract

We have investigating the use of *Deep Learning* (DL) to process sequences of images captured by satellites aiming to improve the quality of river flow forecasting methods, because current ones are still not accurate enough to support efficient management of large national electrical systems. Towards this goal, we are assessing the accuracy of DL networks in extracting information from image sequences by means of classification processes. We have set a test environment composed by an image sequence generator, some generating models, the Nvidia DIGITS tool, two DL preset networks, and the needed hardware. Each model produced one image sequence and one data series corresponding to a selected measure in the images. We have trained the DL networks and evaluated its accuracy in extracting the measured data. In this paper, we show that the performance of DL is extremely sensible to the image type, the measure taken into account, and the DL network applied. Our process presented better performance recognizing coverage area rates in images that resemble clouds and linear distances, but had poor accuracy with angles.

**Keywords:** Deep learning, synthetic images, image classification, river flow forecasting.

---

### 1. Introduction

The river inflow forecast is needed to plan and operate electrical systems that depend on hydroelectric plants. The more accurate the forecasting, the better the system efficiency. There are several methods to execute this task, but their precision still must be improved, as indicated in [1]. In a previous work ([2]), we applied a deterministic model for hydrological simulation [3], compared our results to the current models used by companies [4], and showed the better performance of that model.

However, there is still room for improvements. We have found works driven by Artificial Neural Networks (ANNs), as [5, 6, 7, 8, 9], as well as those driven by satellite and radar imagery [10, 11]. The combination of both methods was considered a valid and promising approach by [12, 13, 14].

An attempt to employ DL and images in solar flare forecast can be found in [15], that lists two advantages of DL in their context: a) DL can accept much large raw data as direct input; b) DL algorithms are *online learning* algorithms, i.e., they can incrementally learn as new data items become available, what is a desirable feature for real-time forecasts.

However the feasibility and the accuracy upper bound of the combination of image from satellites and historical data series was not demonstrated at all for the river flow forecast.

As we stated in [16], the satellite imagery and historical data are too complex to the first evaluation of the DL algorithms. The real images we have received from *Geostationary Operational Environmental Satellite* (GOES) are 1,870 x 1,714 pixels sized, while several works use 256 x 256 pixels images.

---

<sup>1</sup>E-mail Corresponding Author: [andre@ft.unicamp.br](mailto:andre@ft.unicamp.br)

For instance, the ImageNet dataset is a collection of more than 15 million labeled high-resolution images that is used in world contests to push the state-of-art limits of image recognition, according to [15]. The same authors have down-sampled the images from that dataset to 256 x 256 pixels and processed them with DL in 224 x 224 pixels in order to do their research.

So, we have decided use synthesized images to explore the DL limits, as [17] has demonstrated this method in the training of ANNs and [18] pleads it is a cheaper solution for the production of large training sets. Moreover, we have found in [19] motion detection results in pictures that resemble those generated by our models.

In briefly, we generated image sequences associated with a given scalar measure, trained two networks with them, and tried to recover the measures from a set of unlabeled images subsequently submitted to DL. The DL tool itself collected the statistics we used to assess the results.

We have found that the process is very sensible to the characteristics of the images, the target measurement, and the network configuration. However, we reached Top-1 and Top-5 accuracies higher than 75% recovering coverage area of images that resembles actual clouds. Furthermore, the distinct networks have a non-linear, non-correlated, and hard-to-predict levels of accuracy. The next sections present our methods in details, our experimental results, and their discussion.

## 2. Models and Tools

The synthetic images were produced by seven basic models that define the patterns embedded in each image set. An image sequence must be viewed as an *animation* as one sees a cartoon on TV. Table 1 briefly describes the models. Three models define a small square wandering on the image, according to different walking patterns, and they have a variant where a reference line (rl) is drawn linking the center of image to the center of the square. Because the patterns are embedded *in the sequence*, there is no reason to compare single images among the models, since some images produced by one model are identical to those generated by other model. Figure 1 presents samples of distinct images used in this work. For now on, we consider *ten models* for simplicity, summing the seven basic ones and the three variants.

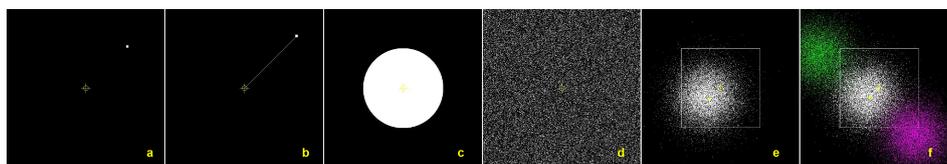


Figure 1:

Samples of images - a) Moving Point, Random Point, and Rotating Point; b) Moving Point, Random Point, and Rotating Point with the reference line; c) Flooding Area; d) Splashing Area; e) Gaussian Cloud; f) Triple Cloud.

We have developed a Java program, named **Fisgen - Fast Image Sequence Generator**, that creates image sets following those defined models and some user choices, as set size, image resolution, image file type, grouping factor, and noise level [16]. There is an option to reproduce the temporal mismatch between observed and predicted data. The final version of Fisgen will be licensed by the Agency of Innovation of Unicamp<sup>2</sup>, possibly as an open source program.

For each model, Fisgen has created a set of images grouped along a series of folders according to the measures that need to be retrieved, complying with the *Nvidia Deep Learning GPU Training System (DIGITS)* version 6 input format.

For instance, the *Moving Point Model* produces an image sequence that, if it is seen as a single animation, shows a small square crossing diagonally the image area. This image sequence is distributed in a set of folders according to the euclidean distance between the center of the moving square and the center of the image. In

<sup>2</sup>Inova - Agência de Inovação da Unicamp: [www.inova.unicamp.br](http://www.inova.unicamp.br)

Table 1: Description of the models used to generate image sequences and associated measures used in tests

Model	The sequence as an animation where...	Measures
Moving Point (MP)	A small square crosses diagonally the image from the left to the right	Euclidean distance from the image center
Random Point (RA)	A small square appears randomly in the image	Euclidean distance from the image center
Rotating Point (RP)	A small square walks along a non-drawn fixed circular path	Angle of arc
Flooding Area (FA)	A centralized circle grows	Circle radius
Splashing Area (SA)	A set of points is randomly distributed in the image	Coverage rate
Gaussian Cloud (GC)	A set of points suggesting a cloud crosses diagonally the image	Coverage rate of a fixed centralized rectangle
Triple Cloud (TC)	Three GC instances suggesting an aggregate of clouds crosses diagonally the image	Coverage rate of a fixed centralized rectangle

the *Flooding Area Model*, an increasing circle gradually fills the image and the sequence is grouped in folders by the circle radius. In summary, for each measure, one set of folders was recorded.

In each test, one image sequence and its associated scalar data were generated. We have choose only a single measure for each model under evaluation and the set of image files was grouped in 8 folders by the respective measures values. The folder names provide DIGITS the range of values inside the group. From this point, the DL tool was able to get the information needed to be trained, towards a classification process.

After the training phase, we have submitted the *same* image sets as input do DIGITS, but without the group structure, labels, or any clue about their contents. So, we broke the convention cited by [15] of using half of the images for training and half for testing. We have proceeded in this way because we were looking for the accuracy upper bound of DL. Then, using the same images to train DL and to retrieve the measures, we provided that no new images were presented to the software, avoiding interference of possible new patterns.

The ten models were tested in two DL networks: *Alexnet* (AN) and *Googlenet* (GN). In order to use the standard configurations of both networks, we used 256 x 256 pixels images. We have ran the tests in two conditions: with and without noise. In the first case, the images were generated following the respective models algorithm. In the second case, we used the rate of 5% of noise, what means that Fisgen was instructed to replace randomly about 5% of the total image pixels with the foreground colors, i.e., the color used to draw the element to be considered by DL. The tests have employed 10,000 synthetic images for each model.

All tests were done without temporal displacement and the parameter *epoch* of the DIGITS was set to 200. For each test scenario, the following results were collected: *Top-1*, *Top-5*, and *Per-class Accuracy*. We have used one Nvidia Titan Xp GPU<sup>3</sup> with the following specifications: 3,840 CUDA Cores, Boost Clock 1,582 MHz, 12 GB GDDR5X 11.4 Gbps.

### 3. Results and Discussion

The Top-1 accuracy was always smaller than Top-5 accuracy. The DL tool Top-1 accuracy ranged from 12.7% to 99.5% (two isolated cases) and Top-5 accuracy ranged from about 55% to 100%. Table 2 shows the results of all tests and allows performance comparisons.

As a general rule, the introduction of some noise in the images reduced the performance of the DL, with few exceptions. Top-5 accuracy was 100% in 13 out 20 noiseless scenarios and 6 (plus two 99.9% scores) out 20 noisy cases.

Both network configurations are somewhat similar in the accuracy range comparing the minimum and maximum values for Top-1 and Top-5 groups, independently of noise. Average and median indicate that, for the overall set of scenarios, AN has a better accuracy in most cases without noise, whereas GN performs better with noise.

<sup>3</sup>Full specifications in [www.nvidia.com/en-us/titan/titan-xp/](http://www.nvidia.com/en-us/titan/titan-xp/)

Table 2: Results of the processing - 256 x 256 pixels images

Model	Alexnet				Googlenet			
	0% Noise		5% Noise		0% Noise		5% Noise	
	Top 1	Top 5	Top 1	Top 5	Top 1	Top 5	Top 1	Top 5
MP	62.9	100.0	14.0	70.0	56.5	100.0	52.5	99.4
MP rl	58.3	100.0	13.8	69.2	63.2	100.0	14.0	70.0
RA	66.6	100.0	51.6	100.0	27.6	87.3	26.6	90.7
RA rl	64.2	100.0	52.8	100.0	63.0	100.0	53.0	99.9
RP	12.9	64.5	12.9	55.4	12.9	64.5	12.9	64.5
RP rl	13.2	64.9	12.9	64.4	12.9	64.5	12.9	64.5
FA	61.3	100.0	13.8	69.1	57.8	97.0	57.0	99.0
SA	12.7	63.6	12.7	63.6	51.3	100.0	63.8	100.0
GC	99.5	100.0	99.5	100.0	79.4	100.0	99.5	99.9
TC	84.4	100.0	75.4	100.0	79.4	100.0	76.2	100.0
Maximum	99.5	100.0	99.5	100.0	79.4	100.0	99.5	100.0
Minimum	12.7	63.6	12.7	55.4	12.9	64.5	12.9	64.5
Average	53.6	89.3	36.0	79.2	50.4	91.3	46.9	88.8
Median	62.1	100.0	13.9	69.6	57.1	100.0	52.8	99.2

In presence of noise, AN reduced its accuracy in the most cases and kept the indicator in the remaining ones. In the same way, noise decreases the GN accuracy in most cases and some scenarios were kept unchanged, but there are 4 cases of accuracy increase.

Taking each model into account, we observed plenty of performance scores in the tests.

Moving Point and Random Point image sequences were correlated with the euclidean distance between the image center and the small walking square center. The accuracy of both networks decayed with noise for Linear Moving Point, GN having better result in most cases. For Random Point Model, we observed some differences: AN kept the 100% Top-5 accuracy with and without noise; GN had a poor performance without a reference line.

Each image generated by Rotating Point was related with the angle of the arc described by the small square in its circular path. Both networks were unable to identify the groups, presenting accuracy about 13% for Top-1 groups and less than 65% for Top-5. Noise and reference line do not appear to play an important rule in this case.

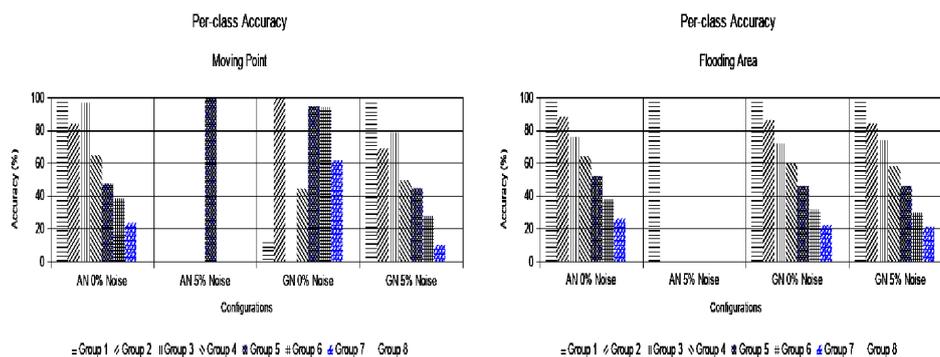


Figure 2:

Per-Class Accuracy - Moving Point (left) and Flooding Area models (right)

Flooding Area model produces image labeled by the radius of a centralized growing circle. AN showed a slight better accuracy in noiseless scenarios, but decayed when noise was introduced. On the other hand,

Table 3: Pearson Correlation Coefficient for Per-Class Accuracy data series

	AN 0% Noise	AN 5% Noise	GN 0% Noise	GN 5% Noise
<b>Moving Point</b>				
AN 0% Noise		-0.11	-0.10	0.98
AN 5% Noise	-0.11		0.41	-0.03
GN 0% Noise	-0.10	0.41		-0.14
GN 5% Noise	0.98	-0.03	-0.14	
<b>Flooding Area</b>				
AN 0% Noise		0.54	1.00	1.00
AN 5% Noise	0.54		0.57	0.58
GN 0% Noise	1.00	0.57		1.00
GN 5% Noise	1.00	0.58	1.00	

GN was immune to the random noise.

The relative number of drawn pixels, or coverage rate, was measured for each image generated by Splashing Area. In this case, AN was indifferent to the noise, but presented low accuracy, whereas GN reached better results and performed better in noisy images.

Gaussian Cloud and Triple Cloud models associate each image with the area of a centralized rectangle covered by the synthetic clouds along their path in the draw. AN had a better Top-1 accuracy without noise; GN had a performance increase when noise was in place for Gaussian Cloud. Both networks have similar results for Top-5 accuracy, with or without noise, reaching 100%.

The *Per-Class Accuracy* recorded by DIGITS in the tests showed an irregular pattern, because some of runs performed better for the first groups, others for the last groups, and others have peaks in random groups. Figure 2 presents the per-class accuracy for Moving Point and Flooding Area. Illustrating this irregular behavior, in the first case, the patterns among the 4 tests are quite different, but, in the second case, 3 out 4 patterns are similar, whereas one is completely different.

Table 3 lists the Pearson Correlation Coefficient calculated for the Per-Class Accuracy data series showed in Figure 2. In the first case, the absolute values span from 0.03 (no correlation at all) to 0.98 (almost perfect match) and there are negative coefficients. The *smallest* absolute correlation was verified between the per-class accuracy of AN with and without noise, i.e., the accuracy of the network could not be correlated with itself when the noise was changed. Conversely, the greatest absolute correlation was between AN *without* noise and GN *with* noise: the two networks, each one in one opposite scenario. Except for this last condition, no correlation among the data could be considered.

In the second case, the situation is quite different. There are no negative coefficients and their values span from 0.54 to 1.00. The perfect correlation was verified between AN without noise and each of GN runs, and between both GN runs. We have observed the same overall irregular behavior in the other tests.

#### 4. Conclusions

We are investigating the use of DL extracting information from image sequences, aiming to improve river flow forecasting methods. Because the feasibility of it was not still demonstrated, we are conducting an exploratory research to evaluate the accuracy of DL in data extraction. In the current phase, we have used synthetic image sequences, generated by simple models, to keep all the image parameters under strict control. Our experimental tests were carried out exercising standard configurations of AN and GN from Nvidia DIGITS running over Titan Xp GPUs. We have used 256 x 256 pixels images, in sequences of 10,000 individual images.

We have concluded that the process we are doing is extremely sensible to several factors: the image type, i.e., what is drawn in the picture; the measure taken into account, because DL had different accuracies

according to the underlying data of the images; and, the DL network employed to do data extraction, since the performance has varied in case by case basis.

Our DL tool showed promising results extracting data from GC and TC models, that are closer to actual clouds than the others, but was not suitable to recover measures in some situations, particularly for RP, and RP + rl models. So, DL had better results extracting coverage rates and linear distances, but was inefficient to recognize arc angles. Coverage area of the entire image was better recovered by GN processing SA model.

A finer look in the *per-class* accuracy of DL has revealed very irregular, non-correlated, and non-linear patterns in performance. We intend to make some adjustments in the DL parameters, in order to reach superior Top-1 accuracy over our tested scenarios.

**Acknowledgments.** We gratefully acknowledge the support of **NVIDIA Corporation** with the donation of the Titan Xp GPU used for this research.

## References

- [1] L. G. F. Guilhon, V. F. Rocha And J. C. Moreira, Comparison of natural inflows forecasting methods to hydroelectric plants, *Revista Brasileira de Recursos Hídricos*, 12(3), 13-20, (2007).
- [2] T. Rocha, I. G. Hidalgo, A. F. Angelis and J. E. G. Lopes, Hydrological simulation in the Tietê Basin. 4th IAHR Europe Congress, Liege Belgium, (2016).
- [3] J. E. G. Lopes, B. P. F. Braga and J. G. L. Conejo, SMAP: A simplified hydrologic model, *Water Resources Publication (Org.)*, Applied Modeling in Catchment Hydrology, Hittleton, 167-176, (1982).
- [4] M. E. P. Maceira, J. M. Damazio, A. O. Ghirardi and H. M. Dantas, Periodic ARMA models applied to weekly streamflow forecasts, *Int. Conf. on Electric Power Engineering*, Budapest, Hungary, (1999).
- [5] M. K. Akhtar, G. A. Corzo1, S. J. van Andel and A. Jonoski, River flow forecasting with artificial neural networks using satellite observed precipitation pre-processed with flow length and travel time information: case study of the Ganges river basin, *Hydrol. Earth Syst. Sci.*, 13, 1607?1618, (2009).
- [6] T. L. Dias, M. Cataldi and V. H. Ferreira, Aplicacao de técnicas de redes neurais e modelagem atmosférica para elaboracao de previsoes de vazao na Bacia do Rio Grande (MG), *Eng Sanit Ambient*, v.22 n.1, 169-178, (2017).
- [7] C. F. L. Gomes, S. M. G. L. Montenegro and M. J. S. Valença, Modelo baseado na técnica de redes neurais para previsão de vazões na bacia do Rio São Francisco, *Revista Brasileira de Recursos Hídricos*, Vol. 15, no. 1, (2010).
- [8] D. I. F. Grimes and M. Diop, Satellite-based rainfall estimation for river flow forecasting in Africa. I: Rainfall estimates and hydrological forecasts, *Hydrological Sciences - Journal des Sciences Hydrologiques*, 48(4), (2003).
- [9] W. S. Souza and F. A. S. Souza, Rede neural artificial aplicada à previsão de vazão da bacia hidrográfica do rio Piancó. *Revista Brasileira de Engenharia Agrícola e Ambiental*, Vol. 14, no. 2, (2010).
- [10] F. A. Hirpa, T. M. Hopson, T. De Groeve, G. R. Brakenridge, M. Gebremichael and P. J. Restrepo, Upstream satellite remote sensing for river discharge forecasting: Application to major rivers in South Asia, *Remote Sensing of Environment* 131, 140?151, (2013).
- [11] S. Saha et al, The NCEP Climate Forecast System Reanalysis, *BAMS - Bulletin of the American Meteorological Society*, (2010), DOI: [dx.doi.org/10.1175/2010BAMS3001.1](https://doi.org/10.1175/2010BAMS3001.1).
- [12] M. Egmont-Petersen, D. de Ridder and H. Handels, Image processing with neural networks : a review, *Pattern Recognition*, V. 35, N. 10, 2279 - 2301, (2002), DOI: [http://dx.doi.org/10.1016/S0031-3203\(01\)00178-9](http://dx.doi.org/10.1016/S0031-3203(01)00178-9).

- [13] C. Li, Y. Bai and B. Zeng, Deep Feature Learning Architectures for Daily Reservoir Inflow Forecasting, *Water Resources Management*, Vol. 30, no. 14, 5145-5161, (2016).
- [14] J. Schmidhuber, Review - Deep learning in neural networks: An overview, *Neural Networks*, V. 61, 85 - 117, (2015), DOI: <https://doi.org/10.1016/j.neunet.2014.09.003>.
- [15] Y. Hada-Muranushi, T. Muranushi, A. Asai, D. Okanohara, R. Raymond, G. Watanabe, S. Nemoto, K. Shibata, A Deep-Learning Approach for Operation of an Automated Realtime Flare Forecast, arXiv:1606.01587. (2016).
- [16] Rocha, T.; Hidalgo, I. G.; Angelis, A. F. Synthetic Images Generation for Deep Learning Assessment Towards the Infow Forecast for Power Systems Operation. *DINCON 2017*, October/November 2017.
- [17] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox and J. Clune, Synthesizing the preferred inputs for neurons in neural networks via deep generator networks, arXiv:1605.09304v5, (2016).
- [18] Apple Inc., Improving the Realism of Synthetic Images, *Machine Learning Journal*, Vol. 1, Issue 1, <https://machinelearning.apple.com/2017/07/07/GAN.html>, (2017)
- [19] C. Wöhler and J. K. Anlauf, An Adaptable Time-Delay Neural-Network Algorithm for Image Sequence Analysis, *IEEE TRANSACTIONS ON NEURAL NETWORKS*, VOL. 10, NO. 6, 1531-1536, (1999).

©Author(s) 2020. This work is distributed under the Creative Commons Attribution 4.0 License.

