https://doi.org/10.2352/CIC.2022.30.1.18

CNN color demosaicking generalizes for any CFA

Lise Yannick Bourrier[1], Martial Mermillod[1], Marina Reyboz[2], and David Alleysson[1];
[1]LPNC, Univ. Grenoble Alpes, Univ. Savoie Mont Blanc, CNRS, 38000, Grenoble, France.
[2]Univ. Grenoble Alpes, CEA, LIST, 17 Av. des Martyrs, Grenoble, 38000, France.

Abstract

A convolutional neural network is trained in auto/hetero-associative mode for reconstructing RGB components from a randomly mosaicked color image. The trained network was shown to perform equally well when images are sampled periodically or with a different random mosaic. Therefore, this model is able to generalize on every type of color filter array. We attribute this property of universal demosaicking to the network learning the statistical structure of color images independently of the mosaic pattern arrangement.

Introduction

For reducing cost and cumbersomeness, most digital cameras use a sensor covered by a Color Filter Array (CFA). Sensors capture only one out of the three RGB color components per pixel, and complete images with three colors per pixels are reconstructed numerically by a process called demosaicking. The color associated with captured pixel at a given location is determined by the arrangement of color filters in the CFA.

Most of these CFAs are periodical patterns of small periodic size, repeated through the entirety of the sensor area. The most common one being the Bayer CFA [1], which consists of a 2x2 array having twice the number of green, than red and blue pixels. Random CFAs, i.e., arrays where the color filter at any location of the captured image is chosen randomly, are less explored. Yet they could theoretically trigger less reconstruction artifacts because reconstruction errors are non-periodic [2, 3]. But this imposes associated demosaicking algorithm that can bypass the higher complexity of dealing with several different neighbourhoods (in terms of pixel configuration). Indeed, one issue is that random CFAs lack the statistical regularities exploited by the reconstruction algorithms for periodical patterns.

The goal of our work is to use the generalization capabilities of convolutional neural networks (CNN) to demosaic images filtered with several random CFAs. Our hypothesis is that a CNN trained on a random CFA, lacking the regularities in the pattern, should instead focus on natural image statistics to reconstruct the output of different random CFA. We expect this training to have interesting generalization capabilities, i.e., a CNN trained as such should be able to demosaic any kind of CFA pattern, periodical or not.

Material and method

The architecture of our network called RandColDem was built specifically to address the inverse problem of demosaicing a completely random CFA pattern. Fig 1. gives a view of the architecture of the network. It can be seen as a hetero-encoder, meaning that its main goal is to reconstruct a small portion of a picture from its mosaiced counterpart and an empirically chosen neighborhood size in order to compensate for the large amount of missing information (i.e. as with other demosaicing problems, two-thirds of the information are missing).

We used 16x16x3 image patches as input (image pixels are taken as they are without any linearization), and 2x2x3 output window. The model received patches of mosaicked images and reconstructed tiny super pixels (2x2x3) with the ground truth being patches of the RGB images. Images were completely reconstructed post-CNN, by replacing the different 2x2 output values of the network to their original place in the picture.

For the input, as two-thirds of the information is missing, two of the three pixel values are set at 0. We used a three-dimensional tensor as input, as was done in similar simulations [4]. This allows us to fully reconstruct an image within a single network. Another way to do so could have been to first separately reconstruct the image for each color channel then merge the results similarly to [5] which might have been of help to improve the network overall performance. Our goal was simply to study the pros and cons of random CFA vs periodical ones, so the simpler method was preferred.

Two hetero encoders, both with the exact same structures, were tested. One is built to reconstruct images mosaicked through fully random CFAs. The other reconstructs images which were mosaicked using a periodical CFA pattern, to provide a basis for comparison and analysis of RandColDems' inner reconstructing patterns. Overall, both networks are designed with an encoder part, a bottleneck part, and a decoder part. The three parts are built as follows:

- The encoder part consists of six stacks of convolution layers using ReLU (Rectified Linear units) activation functions and batch normalization layers. The first layer consists of 9x9x32 kernels. The following five layers are all 3x3 kernels with a progressively growing number of filters (32 – 64 – 64 – 128 – 128) and a stride of one or two to progressively reduce the reconstructed patches' size from 16x16 to 2x2.
- The bottleneck part takes as input the flattened 512 unit vectors of the last encoder convolutional layer. The bottleneck squeezes these vectors to 210 values before raising its size again to 512 units. The value 210 was chosen empirically.
- The decoder part reconstructs super-pixels from the output of the bottleneck part. It is constituted of 2 deconvolution layers with ReLU / batch normalization layers, which progressively reduce the number of filters (i.e.: dimensionality) until the 2x2x3 output.

The mean-squared error loss-function was used to train the network. We used the Adam optimizer [6] with a learning rate parameter set to 1e-3. Training was stopped when validation error stopped improving after five epochs, which took approximately 2 hours with our GeForce 2080 RTX GPU.

Dataset

The dataset was built from 180 pictures chosen from Flickr (selection criterion was picture with balanced hue content, saturated and good looking). As test sets, we used the MCM [7] and the KODIM [8] datasets. No image from the KODIM dataset was used when this dataset was the one tested, similarly for MCM.

Since the reconstructed outputs are tiny, this small number of images becomes a large dataset. For each train/validation procedure, we randomly selected about 2 million 2x2 “center” and 16x16 “neighborhood” patches. To do so, each image was sampled into many 2x2x3 super pixels plus their corresponding 16x16x3 neighborhood. Therefore, there were enough images patches to perform training within reasonable time delays on a recent personal computer.

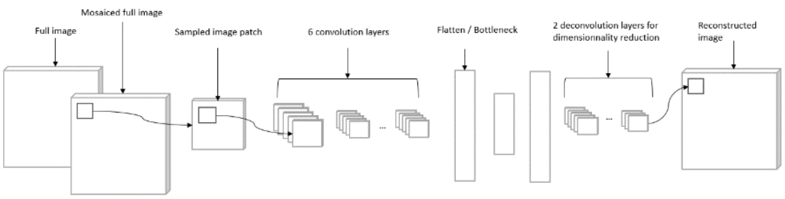


***Figure 1** Schematic view of RandColDem. The network demosaics tiny image patches using neighbourhood information. The same architecture can be used either for demosaicking periodical or random CFAs.*

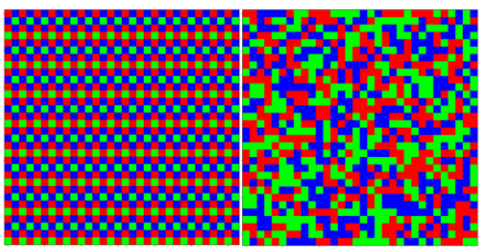


***Figure 2**. Examples of the CFAs used for (left) training the periodical model and (right) training the fully random model. For ease of visualization we show only 32x32 pixels, but every pixel of the random mosaic is picked randomly (i.e. for any image, each pixel is randomly chosen between red, green, and blue).*

Fig.2 showcases two examples of CFA patterns which were used to create the mosaicked images. All images in the database are mosaicked with one of these patterns. For the random model, the CFA pattern was a fully random trichromatic mosaic of the size of the input images, which means each pixel was randomly attributed to either the red, green, or blue channel, and set to 0 for the other two. No statistical property of randomness, e.g., equalizing the number of colors per patch or limiting the conglomeration of a particular color, was added. The mosaic was applied on each image of the dataset by element-wise multiplication of each color channel with the associated CFA.

For experiment 1, the periodical model, the 4x4#2 mosaic was used. This mosaic was proposed by Amba & al. [3] as an alternative to the more traditional Bayer pattern. We chose this pattern for our periodic model as it was the one which provided the best performances after training. During experiment 2, we also used the Bayer CFA to mosaic the image and evaluate the capabilities of both our models to generalize to another kind of periodically mosaicked image.

In experiment 2, our objective is to confirm that the random network has indeed learned to demosaic under an approximation of luminance, and independently from the kind CFA. To do so, we test the capabilities of both the periodic and random model to perform demosaicking of images filtered through CFAs that are different from the one the models have been trained on. In the case of the periodic model, demosaicking is tested using (1) images filtered with a fully random CFA and (2) images filtered with the Bayer CFA. In the case of the random model, it is tested using (1) images filtered with the 4x4 CFA (i.e. the one used to teach the periodic model) (2) images filtered with the Bayer CFA and (3) images filtered with another fully random pattern. Figure 3 shows examples of reconstructed images for the two models.

Results

Experiment 1

In this first experiment, we provide the results of RanColDem when demosaicking random CFA and periodic 4x4-2 CFA, with quality measured by the widely used Peak Signal to Noise Ratio (PSNR) metric. We then focus on the differences between how both models function. Our analysis concludes that the periodical and the random model proceed in a completely different ways, more specifically with regards to how luminance is handled. We highlight this by analysing the results of a post-training degradation of the CFA mosaic’s colors.

Table 1 shows the results for the KODIM and the MCM datasets. The periodical model reaches slightly higher values (about 2 points PSNR) but the differences between both models in similar conditions is small. For the two models, reconstruction quality is good, despite a relatively low PSNR, as confirmed by the visual evaluation provided in figure 3.

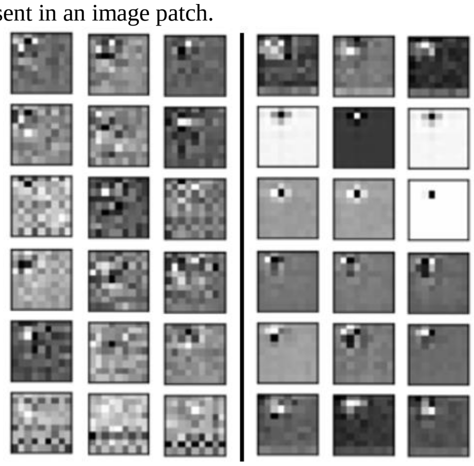
***Figure 3**. Reconstructed KODIM image 19. For both the periodical and Random RandColDem, reconstruction quality is good. “Traditional” visual artifacts on this image, such as aliasing in the fences, are absent. Some colors are slightly modified (i.e. reddish taint on the fences, even more so for the periodical model). The random model picture is slightly more blurred. Overall colors seem better represented on the random model whereas blurriness is less pronounced on the periodical one.*

It should be noted that these results are obtained using mosaicked images as input and no post-processing once the image patches have been reconstructed. Raw performance can be improved by adding mechanisms such as providing the exact pixel values of the available pixels (i.e. the ones which are not changed during the mosaicking process) to the output or rebuild the residual of the mosaicked pictures and fully reconstruct the desired patches post neural network. PSNR can also be improved by simply augmenting the number of kernels at each convolution layer. A third way to increase PSNR would be to provide a more carefully crafted dataset of images as input. These implementations would also raise the PSNR to values closer to the best demosaicking models (i.e. DMCNN-VD [9], whose values for both KODIM and MCM are provided in Table 1), at the cost of simplicity. Performance is not our main objective, rather, obtaining an equal baseline to compare and analyze the demosaicking process of periodical and fully random models is.

Model	Test Dataset	
	KODIM	MCM
RandColDem - Random CFA	36,9	33,3
RandColDem - 4x4#2 CFA	39,2	35,5
DMCNN	40,4	36,6
DMCNN-VD	43,4	39,5

Table 1. Performance (PSNR) of both the periodical model and random RandColDem model for the KODIM and MCM datasets. DMCNN-VD and its shallow version, DMCNN, are provided for comparison [9]

Both networks indeed learn the most efficient way to achieve the required task. For the periodical model, the better approximation to reconstruct the image is to simply guess which color is missing out of the repetitive pattern which is present both at the central super-pixel location and at the neighborhood, so the pattern can be deciphered without need to focus on the image statistics. For the random model, since the neighborhood lacks completely random pixels, the better approach is to grasp luminance instead and therefore focus on the different kinds of shades present in an image patch.



(a) | (b)

***Figure 6**. Examples of 18 out of the 32 learned kernels for the first 9x9 convolution layers of periodical model (a) and the random model (b). We can see that the periodicity of the mosaic is visible in the filters of the periodical model, while the random model does not show the same redundant pattern. Rather, it is much more focused on a central element and much more sensitive to luminance elements than the periodical model.*

At a pure performance level and without demosaic-specific optimizations to the models, the task conducted by the periodical model is more efficient in terms of metrics, probably because it is simpler and does not need to spend as much attention on image statistics, more specifically light. This however comes with a price; Figure 7 shows what happens when a portion of a single color in the mosaic is destroyed. Generally, as the level of degradation increases, colors are more preserved on the random model. For this model, destroyed pixels mostly appear as dark grey or seemingly random colors, while artifacts which are extremely specific to the destroyed color (i.e. magenta when green is lost, turquoise when red is lost, yellow when blue is lost) instantly appear in the periodical model. Those specific color artifacts also end-up appearing for the random model when the degradation factor augments, but to a lesser degree.

***Figure 7**. Examples of reconstructed images with degraded mosaics. For the simulation a proportion of randomly chosen pixel are set to zero. The destructed mosaic pixels can be seen in both the random model and the periodical model. However, in the case of the periodical model, color is more false: magenta artifacts appear much faster when green is degraded and similarly with cyan artifacts when red destroyed. The same result happens with blue destroyed and yellow artifacts (not shown here to save place).*

Experiment 2

Note that since both network structures are the same, these differences in reconstruction behavior can only come from their input during training. In other words, when tuned for a demosaicking task, a neural network will, in the case of a periodical pattern, learn to fill-in the periodically missing colors, and in the case of a random pattern, the statistics of the images it should reconstruct. Since it is image-based rather than mosaic-based, we assume that this second way of completing demosaicking should come with independence to the kind of CFA used to mosaic an image. Experiment 2 tests this by analyzing the consequences for both models when images built on other CFA patterns are fed to them.

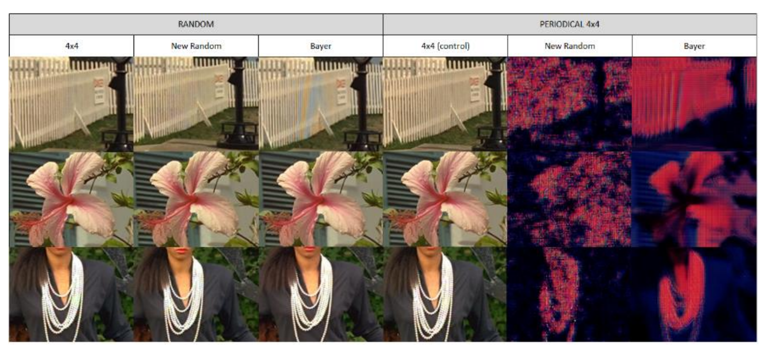


***Figure 8**. Examples of reconstructed images for the random model using different mosaic patterns which were never used for training. Interestingly, the random model performs slightly better (about 0.5 PSNR difference) on the 4x4 CFA that on a random mosaic, whether the one it was trained on or a different one. Conversely, the performances are slightly worse (2 PSNR difference) when the random model demosaics images filtered with the Bayer CFA. The periodical model, however, shows extremely low performances when demosaicking unlearned CFA patterns. In the case where another periodical CFA (i.e.: the Bayer CFA) is used, color information is lost but the shape of*

the picture is conserved. Against a randomly mosaiced image, most of that shape information is also lost.

As can be seen, the random network is able to demosaic all the presented input. This suggests that the network has learned the statistics of the image when trained off a random pattern. Some traditional artifacts appear on the reconstructed image crafted under the basis of the Bayer CFA (i.e. blue/yellow color lines on high frequency areas). Conversely, the periodical model's inability to reconstruct images mosaicked under different patterns than its own confirms it learned the mosaic rather than the image.

Discussion

The main conclusion of the article is that our model is able to learn, at the price of a small reduction of performance of PSNR, a universal function allowing learning and generalizing demosaicking from any type of sensors CFA. This property was obtained with not much artifacts for any type of CFA (periodic or not). The first step of the network is a bank of 32 convolution filters that did 32 different representations of color input as projection on a particular axis (in a 16x16x3 dimensional vector space), given by the learned coefficients present in the 32 convolutional filter. Because those filters mixed color components in the computation of the output, they can be considered as projections on the luminous, achromatic, axis. Orientation of the luminous axis depends on the coefficient of the filters and the particular color arrangement on the input pattern. Once the network is trained, filter coefficients are fixed and determined by the values of pixels of the input. Because the corresponding color of a particular pixel is set randomly, the direction of the luminance projection is not fixed. In the regular case, one can argue that several luminous projections are used and weighted accordingly to enable reconstruction of chrominance. But for the random case, the several luminous projections depend on the input pattern. Contrary to the regular case, the neural network cannot use an estimate of colors based on a multitude of luminous projections on different axis. It is thus likely that the network trained with random input color arrangement learned something on the statistics of image than the configuration of the mosaic itself.

Knowing why the network trained with random arrangement of color samples provides a generalization for any arrangement, even regular, remains an open question. The network has learned several different input patterns despite it being impossible to have trained on all patterns. To be specific, this is impossible because there are $3^{16x16} > 1.3E122$ different possibilities of placing three colors on a 16x16 grid, while our training set has about 2 million elements. We can affirm that the network is able to generalize based on a reduced set of examples. Learning rate is lower in random case than in regular case (50 epochs for the regular net and twice this number for the random case), and the reconstruction quality is lower in random case compared to the regular case (~2dB less for KODIM database). But it remains spectacular that the network can reconstruct raw image having any arrangement of colors. This raises similar questions at the level of human retina that is able to learn the demosaicking process from natural images on the basis of a single random mosaic of photoreceptors (for a specific individual, and different mosaics among different individuals). This will be tested in further studies but we assume here that this property is related to the ability of biological (and artificial) neural networks to capture and generalize the statistics of natural images.

Acknowledgments

This work has been partially supported by MIAI@Grenoble Alpes, (ANR-19-P3IA-0003).

References

[1] Bayer, B. E. (1976). Color imaging array. United States Patent 3,971,065.

[2] Condat, L. (2010). Color filter array design using random patterns with blue noise chromatic spectra. Image and Vision Computing, 28(8), 1196-1202.

[3] Amba, P., Dias, J., & Alleysson, D. (2016, November). Random color filter arrays are better than regular ones. In Color and Imaging Conference (Vol. 2016, No. 1, pp. 294-299). Society for Imaging Science and Technology.

[4] Syu, N. S., Chen, Y. S., & Chuang, Y. Y. (2018). Learning deep convolutional networks for demosaicing. arXiv preprint arXiv:1802.03769.

[5] Tan, R., Zhang, K., Zuo, W., & Zhang, L. (2017, July). Color image demosaicking via deep residual learning. In IEEE Int. Conf. Multimedia and Expo (ICME) (Vol. 2, No. 4, p. 6).

[6] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[7] Kodak Database http://r0k.us/graphics/kodak

[8] Zhang, Lei, et al. "Color demosaicking by local directional interpolation and nonlocal adaptive thresholding." Journal of Electronic Imaging 20.2 (2011): 023016-023016.

[9] Syu, N. S., Chen, Y. S., & Chuang, Y. Y. (2018). Learning deep convolutional networks for demosaicing. arXiv preprint arXiv:1802.03769.