International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.

https://ijpsat.org/
Vol. 55 No. 1 January 2026, pp. 456-463

Optimization Of The Efficient Sub-Pixel Convolutional Network *Model For Satellite Image Super-Resolution: Study Of Epoch And Batch Size Hyperparameters*

[1]Harimino Andriamalala RAJAONARISOA, [2]Adolphe Andriamanga RATIARISON

[1,2]Dynamic laboratory of Atmosphere, Climate and Oceans
Doctoral School of Physics and Applications
University of Antananarivo
Madagascar
[1]hariminondriamalala@gmail.com, [2] adolphe.ratiarison@univ-antananarivo.mg
Corresponding Author: hariminondriamalala@gmail.com

Abstract— This study applied the ESPCN model to the super-resolution of geostationary meteorological satellite images. Using 100 pairs of low- and high-resolution images, the model was trained and tested the by optimizing the epoch and batch-size hyperparameters. The analysis of the PSNR between the reconstructed images and the target images made it possible to identify the optimal values that ensure the best reconstruction quality. The results demonstrated the importance of hyperparameter tuning to improve model performance in the context of satellite image processing.

Keywords— Batch size, Deep learning, Epoch, ESPCN, Image super-resolution, PSNR.

I. INTRODUCTION

Image super-resolution (ISR) is an image-processing technique aimed at reconstructing a high-resolution image from one or more low-resolution images. It has become a major challenge in many fields, particularly in remote sensing, where the quality of satellite imagery determines the accuracy of environmental, meteorological, or climate analyses [1], [2]. In particular, geostationary meteorological satellites produce images with a high temporal resolution but often with a limited spatial resolution. Improving the resolution of these images without compromising their integrity is therefore an essential task [3].

In recent years, the emergence of deep learning and in particular, Convolutional Neural Networks (CNN) has enabled significant advances in the field of image super-resolution, surpassing the performance of traditional interpolation-based approaches [2]. Among recent models, the ESPCN (Efficient Sub-Pixel Convolutional Neural Network) stands out for its ability to produce high-quality images at a low computational cost, making it suitable for real-time applications [4]. In this study, the ESPCN model was applied to a dataset of 100 pairs of satellite images (low and high resolution) obtained from geostationary meteorological satellites. The aim was to optimize the model’s performance by determining the most effective training hyperparameters, in particular the number of epochs and batch size. The model’s performance was evaluated using PSNR (Peak Signal-to-Noise Ratio), a standard indicator of reconstruction quality [5].

International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.

https://ijpsat.org/
Vol. 55 No. 1 January 2026, pp. 456-463

II. MATERIALS AND METHODS

A. Datasets

The study was based on a set of 100 pairs of low-resolution (LR) and high-resolution (HR) satellite images, obtained from the same geostationary meteorological satellite but at two different resolutions.

The low-resolution images were downloaded from the website: https://eumetview.eumetsat.int/static-images/MSG/RGB/NATURALCOLOR/FULLDISC/ and the high-resolution images from: https://eumetview.eumetsat.int/static-images/MSG/RGB/NATURALCOLOR/FULLRESOLUTION/.

These datasets cover various meteorological conditions and time periods, ensuring sufficient representativeness for model training and testing. The images were preprocessed to ensure precise spatial alignment and normalization of pixel values.

B. ESPCN Model

The Efficient Sub-Pixel Convolutional Neural Network (ESPCN) [5] is a convolutional neural network specifically designed for image super-resolution. Unlike traditional methods that interpolate a low-resolution (LR) image before processing it through a network, ESPCN performs the entire reconstruction in low-resolution space, which allows for optimization of both the quality and speed of super-resolution.

The detailed steps of the process used to obtain a super-resolved (HR) image from an LR image using ESPCN are as follows:

a) Step 1 : Input — Low-resolution image

The network takes as input an LR image of size W×H (width × height), with a certain number of channels (for example, 1 channel for grayscale, or 3 for RGB, i.e., a color image).

b) Step 2 : Feature extraction through convolutional layers

The low - resolution image is first processed through several successive convolutional layers. Each layer applies a set of convolutional filters that extract local spatial features (edges, textures, patterns). These layers produce a rich and hierarchical representation of the information contained in the LR image.

In this study, four successive convolutional layers were used, which are summarized in Table 1.

TABLE I. THE DIFFERENT CONVOLUTIONAL LAYERS

Convolutional layer	***Number of filters***	***Kernel size***	***Role***
First Conv2D layer	64	5x5	Extraction of broad features
Second Conv2D layer	64	3x3	Refinement
Third Conv2D layer	32	3x3	Reduction of the number of features
Fourth Conv2D layer	9	3x3	Preparation for spatial rearrangement, or depth-to-space

c) Step 3 : Generation of feature maps

After these convolutional layers, the network produces a feature tensor of dimensions $W \times H \times C \times r^2$, where r^2 is the desired upscaling factor (in our experiment, $r = 3$ which corresponds to tripling the resolution). This tensor contains information ready to be rearranged to reconstruct the High Resolution image.

International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.

https://ijpsat.org/
Vol. 55 No. 1 January 2026, pp. 456-463

d) Step 4 : Sub-pixel convolution operation (pixel shuffle)

The final key step is the sub-pixel convolution layer (or pixel shuffle). This layer rearranges the pixels of the feature tensor by grouping them spatially to increase the spatial resolution.

Specifically, this operation converts the $W \times H \times C \times r^2$ tensor into an image of size $(W \times r) \times (H \times r) \times C$, by redistributing the r^2 extra channels into the spatial domain (neighboring pixels), effectively 'multiplying' the resolution.

This operation is much more efficient than traditional interpolation methods and avoids the information loss associated with conventional techniques.

e) Step 5 : Output — Reconstructed high-resolution image

The final result is a super-resolved (HR) image of size $(W \times r) \times (H \times r)$ with the same channel depth as the original image, containing fine details and improved visual quality.

Figure 1 illustrates the operation of the ESPCN model.

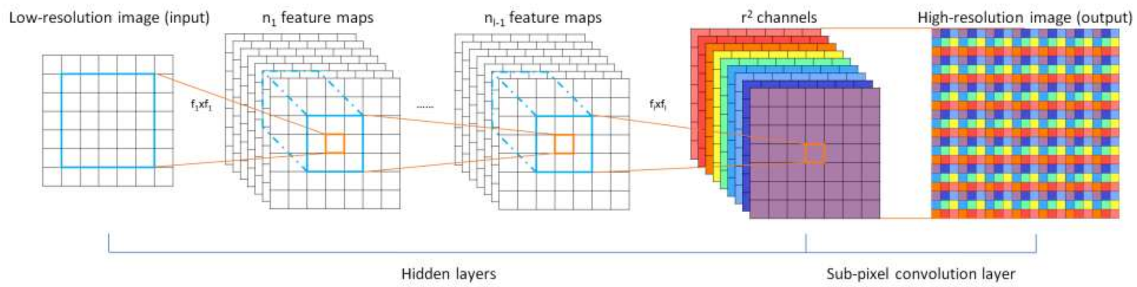

Figure 1: Illustrative diagram of the ESPCN operation (Source: [5])

C. Modeling procedure

The modeling process consists of four steps:

- Step 1: Data preprocessing;
- Step 2: Training the ESPCN model;
- Step 3: Testing the ESPCN model;
- Step 4: Selection of the optimal batch size and epoch parameters.

- **Step 1 : Data preprocessing**

Data preprocessing operations include resizing images and splitting them into datasets for model training and testing.

The original size of images is 800×800. This size is reduced to 400×400 for both low- and high-resolution images. The purpose of resizing the images is to save processing time, allowing the use of the highest possible value for the training parameters.

Furthermore, 73% of the dataset is dedicated to training the model, while the remaining 27% is used for testing. Indeed, model testing is much more objective when the data used for testing are completely separate from the data used for training.

- **Step 2: Training the ESPCN model**

The model was trained using the training data. The training parameters, batch size and epoch, were treated as variables in the modeling. The training was carried out in two steps:

- Set the batch size parameter to 1 and vary the epoch parameter from 1 to 80 in increments of 5. Every 5 epochs, the model takes a single image as input (since the batch size = 1). The convolutional layers then extract the features from the image. The sub-pixel convolution (pixel shuffle) operation then produces the reconstructed image, which is considered to be the predicted high-resolution image. This image was compared with the reference high-resolution image using the PSNR metric.
- Set the epoch parameter to the value corresponding to the maximum PSNR obtained in the previous step, and vary the batch size parameter from 1 to 20 in increments of 1. For each integer batch size between 1 and 20, the ESPCN model was trained in the same way as in the previous step.

- **Step 3: Testing the ESPCN model**

The ESPCN model was tested using the test data. The testing procedure adopted was similar to the training procedure.

- **Step 4: Selection of the optimal batch size and epoch parameters**

The epoch parameter was chosen first. Referring to the curve showing the average PSNR evolution between training and testing as a function of epoch, the optimal epoch value corresponds to the epoch at which the PSNR reaches its maximum. Indeed, higher PSNR values indicate a higher similarity between two images. Since the predicted image was compared with the high-resolution reference image, the maximum PSNR corresponded to the best modeling.

To determine the batch size parameter, the epoch was set to the optimal value determinate previously, and the batch size parameter varied from 1 to 20 in increments of 1. Using the curve of the average PSNR evolution between training and testing as a function of batch size, the optimal batch size value corresponded to the maximum PSNR.

III. RESULTS AND INTERPRETATIONS

A. Result of the search for the optimal epoch

Figure 2 illustrates the progression of PSNR for the training set, test set, and their average over 80 epochs of the ESPCN model with a batch size fixed at 1.

The PSNR starts around 68.5 dB and initially experiences a notable drop until approximately epoch 10, before rapidly rising to a peak around epoch 20 (≈ 69 dB). This behavior reflects a phase of instability typical at the beginning of training: the model strongly adjusts its weights while gradually aligning with the data distribution.

After this first peak, the PSNR gradually decreases, reaching a minimum of around 68.0 dB near epoch 50. From epoch 50 onwards, the PSNR rises consistently, ending up around 68.6–68.8 dB. This late improvement indicates that the model continues to refine its reconstruction, revealing that learning is still active.

The training, test, and mean curves remain extremely close throughout training, indicating the absence of significant overfitting, good generalization ability of the model, and consistency between the training and test distributions.

It was concluded that the maximum PSNR was achieved at the 20th epoch, which is considered the optimal epoch value.

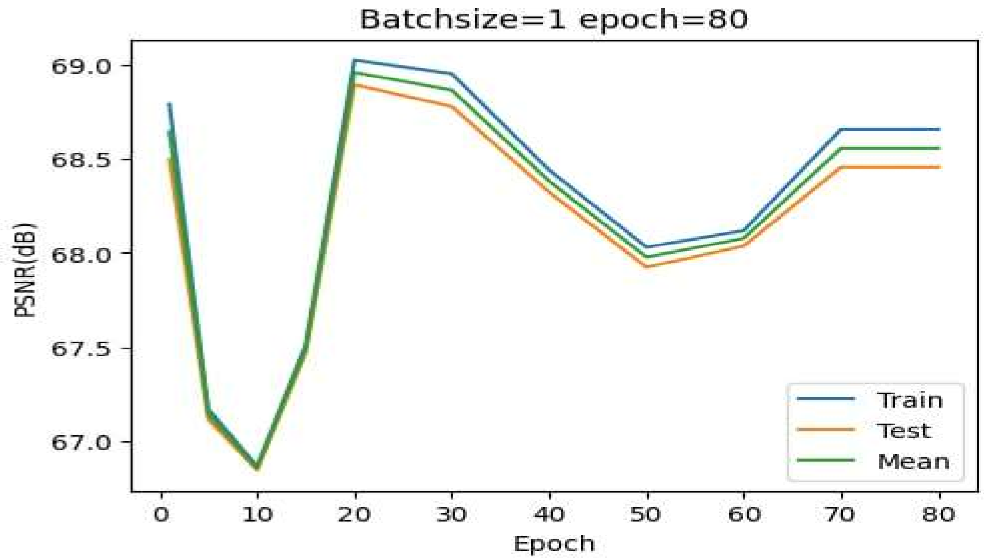

Figure 2: PSNR evolution curves as a function of epoch

B. Result of the search for the optimal batch size

Figure 3 examines the influence of batch size (ranging from 1 to 20) on PSNR after a short training of 20 epochs. For small batch sizes (1–5), PSNR varies significantly, oscillating between 68 and 70 dB. This variability can be attributed to gradient instability when it is estimated from a small number of samples.

The PSNR reaches its maximum (close to 73 dB) when the batch size is between 8 and 10. This indicates that the ESPCN model benefits from an optimal trade-off between gradient stability and data diversity at each update.

The PSNR then drops sharply to around 66 dB for a batch size of 15. Very large batch sizes tend to overly smooth the gradient, reducing the regularizing effect inherent to small batch sizes, and sometimes leading to less efficient learning.

An increase in PSNR is then observed, rising to around 69–70 dB for the largest batch sizes. This shows that beyond a certain threshold, training becomes stable again but without reaching the peak observed for intermediate batch sizes.

As in Figure 2, the training, testing, and mean curves are almost superimposed. The model therefore maintains good generalization regardless of batch size, but the absolute quality of reconstruction varies significantly depending on the training conditions.

Thus, at the end of the experiment, the maximum PSNR was obtained for a batch size of 8, which is considered the optimal batch size.

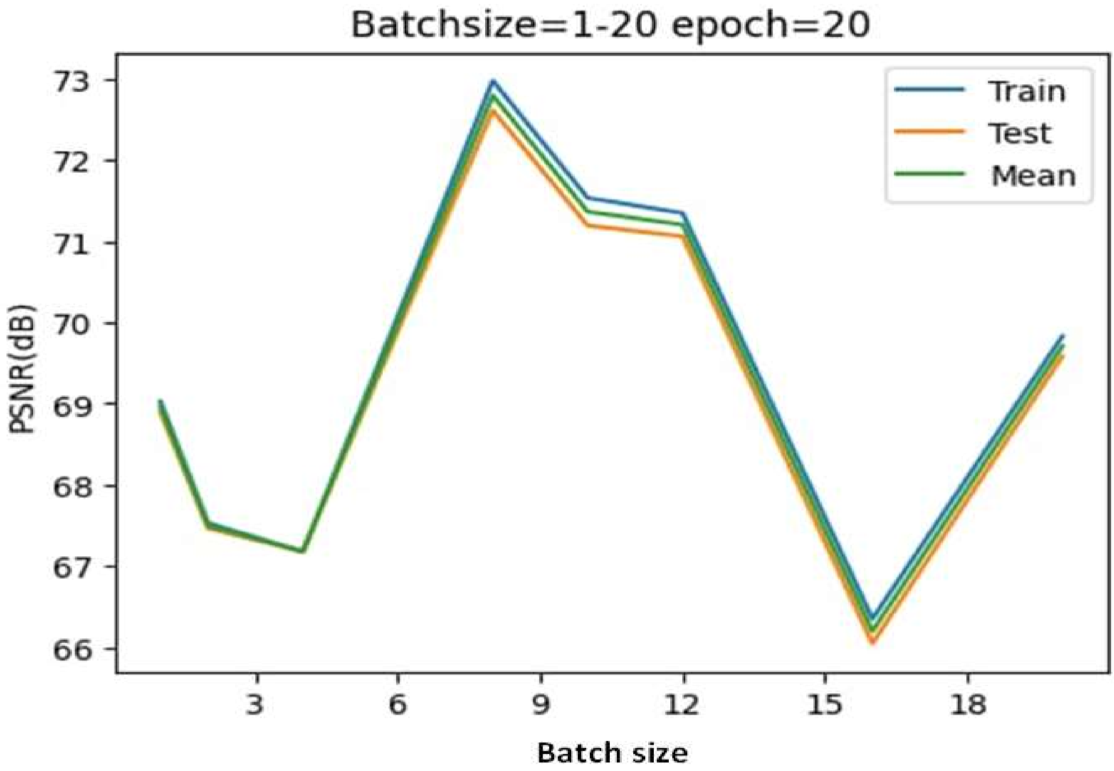

Figure 3: PSNR evolution curve as a function of batch size for a fixed of 20

C. Example of an image super-resolution result with optimal batch size and epoch parameters

Figures 4-a, 4-b, and 4-c respectively show, a high-resolution image, the corresponding low-resolution image (obtained from the same meteorological satellite but at a different resolution), and the image predicted by the ESPCN model using the optimal training parameters (epoch = 20 and batch size = 8). It can be observed that the ESPCN model is able to predict an image that is significantly better than the low-resolution image provided as input.

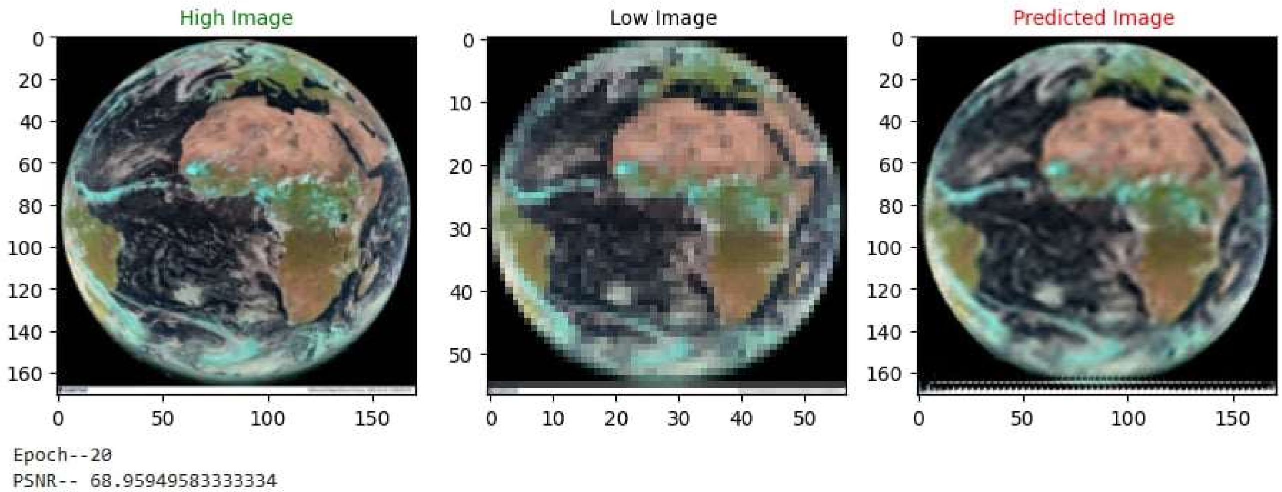

Figure 4-a: High-resolution image; Figure 4-b: Low-resolution image; Figure 4-c: Image predicted by the ESPCN model with optimal parameters (epoch = 20 and batch size = 8)

IV. DISCUSSION

The evolution of PSNR over 80 epochs (Figure 2) first shows an initial phase of instability, followed by a rapid rise to a maximum around the 20th epoch. This behavior aligns with classic observations in deep learning, where the early iterations produce marked fluctuations due to the gradual establishment of a trade-off between the bias and variance of the model [6]. The very small divergence between the training, test, and mean curves suggests good generalization. This stability confirms the robustness of ESPCN against overfitting, as already highlighted in the seminal work by Shi et al. (2016) [5], where the sub-pixel architecture showed a natural compatibility with relatively long training periods without performance degradation.

Epoch 20 was identified as optimal because it corresponded to the point of maximum PSNR. This finding is consistent with other studies where an early optimum is often observed in lightweight architectures designed for super-resolution [7], [8].

The analysis of Figure 3 showed that batch size plays a crucial role in the final reconstruction quality. For small batch sizes, the marked oscillations in PSNR are consistent with the theory that highly noisy gradient updates can disrupt learning at the start of the gradient descent [9], resulting in high intra-training variability.

The optimal value obtained—a batch size of 8—falls within the range generally recommended for compact SR models due to their shallow depth and sensitivity to overly - smoothed gradients [10], [11], [12], [13], [14], [15].

V. CONCLUSION

This work assessed the impact of the number of epochs and batch size on the performance of the ESPCN model applied to image super-resolution, primarily analyzing the evolution of PSNR during training. The results showed that the learning dynamics of the model exhibit an initial phase of instability followed by rapid improvement, peaking at the 20th epoch. This point corresponds to the maximum PSNR observed and represents the optimal epoch for the experimental context studied.

Analysis of the influence of batch size revealed that a precise trade-off between gradient stability and generalization capability was necessary to fully exploit the potential of ESPCN. Although training remained generally stable across all batch sizes tested, maximum performance was achieved with a batch size of 8, confirming that lightweight convolutional models benefit from moderately noisy gradients and more frequent updates.

Overall, the study highlighted that combining a relatively short training duration (≈ 20 epochs) with an intermediate batch size represents an optimal configuration for the ESPCN model in this context. These results are consistent with previous work on super-resolution network optimization and provide practical guidance for hyperparameter selection.

Future work could explore the influence of other factors, such as the learning rate, the choice of loss function, or the use of advanced regularization techniques. Extending the study to deeper or hybrid architectures would also allow for the assessment of the generalization of these conclusions to more complex super-resolution models.

References

[1] Q. Ha, M. Dang, T. D. Hoang, and Y. Kim, "Image super-resolution using deep learning: A comprehensive survey," *International Journal of Automation and Computing*, vol. 16, no. 6, pp. 663–685, 2019, doi: 10.1007/s11633-019-1183-x

[2] Z. Wang, Z. Zeng, Y. Zhang, L. Wang, and L. Zhang, "A comprehensive review of super-resolution in remote sensing: From classical to deep learning era," *Earth-Science Reviews*, vol. 232, p. 104110, 2022, doi: 10.1016/j.earscirev.2022.104110

[3] X. Lu, X. Zhang, Y. Liu, and J. Wu, "Remote sensing image super-resolution via a dual-path network," *Remote Sensing*, vol. 11, no. 13, p. 1588, 2019, doi: 10.3390/rs11131588

[4] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, *et al*., "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 1874–1883, doi: 10.1109/CVPR.2016.207.

[5] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004, doi: 10.1109/TIP.2003.819861

[6] Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016

[7] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2016.

[8] N. Ahn, B. Kang, and K. Sohn, "Fast, accurate, and lightweight super-resolution with cascading residual network," in *Proc. European Conf. Comput. Vis. (ECCV)*, 2018.

[9] S. L. Smith and Q. V. Le, "A Bayesian perspective on generalization and stochastic gradient descent," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2018.

[10] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *Proc. CVPR Workshops*, 2017.

[11] N. Ahn, B. Kang, and K. Sohn, "Fast, accurate, and lightweight super-resolution with cascading residual network," in *Proc. European Conf. Comput. Vis. (ECCV)*, 2018.

[12] S. L. Smith and Q. V. Le, "A Bayesian perspective on generalization and stochastic gradient descent," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2018.

[13] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, *et al*., "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 1874–1883, doi: 10.1109/CVPR.2016.207.

[14] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2016.

[15] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004, doi: 10.1109/TIP.2003.819861.