www.editada.org

International Journal of Combinatorial Optimization Problems and Informatics, 17(2), 2026, 405-415. ISSN: 2007-1558. https://doi.org/10.61467/2007.1558.2026.v17i2.952

Quality and Randomness Assessment of Images Generated by a Convolutional Autoencoder

Rafael Castaneda-Diaz[1, 2], *Daniela López-Betancur**[1], *Carlos Guerrero-Méndez**[1], *Efrén González-Ramírez*[1], *Salvador Gómez-Jiménez*[1] *and Flossi Puma-Tito*[1]

[1] Unidad Académica de Ciencia y Tecnología de la Luz y la Materia, Universidad Autónoma de Zacatecas, Circuito Marie Curie S/N, Parque de Ciencia y Tecnología QUANTUM Ciudad del Conocimiento, 98160 Zacatecas, Zacatecas, México.
[2] Instituto Politécnico Nacional, Unidad Profesional Interdisciplinaria de Ingeniería Campus Zacatecas (UPIIZ), Zacatecas 98160, México.
danielalopez106@uaz.edu.mx, guerrero_mendez@uaz.edu.mx

Abstract. This paper presents an analysis of the quality and randomness of information in images generated with a convolutional autoencoder (CAE). The CAE convolved altered color images from CIFAR-10 dataset. The CIFAR-10 images were altered by randomly setting 30%, 60%, and 90% of pixel values to [0, 0, 0] or [255, 255, 255], respectively. In the validation stage, Mean Square Error (MSE) loss function reached 0.0115 and the accuracy metric 0.7461. Similarity Structural Index Measure (SSIM), Pearson Correlation Coefficient (PCC) and Peak Signal-to-Noise Ratio (PSNR) metrics, assessed quality of generated images. The assessment results ranged as follows: SSIM [0.3251, 0.6830], PCC [0.5034, 0.9358], and PSNR [18.63 dB, 26.04 dB]. The Shannon metric assessed randomness both locally and globally for each image, ranging from 1.25 to 2.49 bits, and from 6.61 to 9.71 bits, respectively. CAE implementations highlight their potential for applications in technological innovation.

Keywords: Autoencoder, Convolutional Neural Network, Similarity Structural Index Measure (SSIM), Pearson Correlation Coefficient, Peak Signal-to-Noise Ratio, Shannon metric.

Article Info
Received March 4, 2025
Accepted Dec 11, 2025

1 Introduction

According to Wang et al. (2004), an objective image quality metric can play two roles in images processing applications: First, it can be used to dynamically monitor and adjust image quality, and the second, it can help to optimize algorithms and parameters settings of image processing system. In this sense, objective assessment of any feature can provide some precise criteria about facts related to the nature of images. In this work we focused on two basic features: quality and randomness of information on generated images with a convolutional autoencoder (CAE). Most of the time, the quality is conceptualized as overall fidelity, clarity, and accuracy in representing visual information (Pappas et al., 2000). The scope of this paper reaches the discussion about assessing quality with objective methods such SSIM, PCC and PSNR, all applied on images generated by a simple proposal of a CAE. The implementation of these metrics was based on several previous works (Wang et al., 2004; Fan et al., 2019; Ilesanmi & Ilesanmi, 2021). On the other hand, randomness of information contained in images refers to the degree of unpredictability of pixel values, i.e., the intensity defined for three values which vary from 0 to 255 per pixel with color images and one value from the $[0, 1]$ interval with grayscale images, respectively. Similarly, in this work for assessing the randomness of the pixel values in the generated images we have used the Shannon entropy metric which measures the amount of uncertainty in the probability distribution on pixel values (Ghojogh et al., 2019; Sparavigna, 2019). In this sense, it is undoubtedly that the process of generating images by a CAE produces new outcome images with useful information.

The initial point in this work is to alter the structural nature of an image. In this case, distortions or noises are considered structural modifications. Noise is defined as a random variation of the values of some pixels, i.e., random variations in intensity or brightness (Venkataraman, 2022). From a subjective human point of view, noise is considered corruption or distortion that results in a low-quality image. It is very common that variational patterns in intensity or brightness resulted from different real applications, e.g., environmental factors for cameras, functionality of sensors, vibrations, dusty scenarios, humidity, chemical compositions, etc. (Fan et al., 2019). In contrast, from a mathematical point of view, distortion could be an opportunity for discovering new facts about images obtained from experimental scenarios or for concluding about the functionality of models and algorithms, e.g., in Lopez-Betancur et al. (2024) state-of-the-art optimization strategies available in PyTorch have been evaluated, using the AlexNet model, pre-trained and coupled with a Multiple Linear Regression (MLR) model for estimating the suspended solids (represented for black pixels randomly distributed on a white background image) in liquid samples.

In the last years, convolutional neural network (CNN) algorithms have received great attention for the flexibility on image denoising image (Ilesanmi & Ilesanmi, 2021). For that reason, based on the architecture and functionality of a CNN, a simple autoencoder (AE) was proposed. Fundamentally, the AE was implemented for compressing and decompressing 50,000 altered images from CIFAR-10 dataset. Through the process of compression and decompression, altered images are gradually cleaned to generate novel images. The dataset of new images can be implemented to make pattern analysis.

The outline of this paper is as follows. In section 2, AE are described in the mathematical sense and their functionality with images. In section 3, the ways datasets were generated and organized are described in detail. Also, the proposal of CAE and its implementation, resources and its performance evaluation are described. Section 4 contains the results of the application of the model as well as the discussions about the implementation of the objective metrics such SSIM index, PCC and PSNR for quality assessment, as well as the results of application of Shannon metric for randomness on information assessment. Finally, section 5 contains some conclusions and suitable work for the future.

2 State of the art

A2.1 Mathematical foundation

The origin of AE dates to the 1980's. The foundation is the learning procedure called back-propagation. This procedure consists of repeated adjustments for weights of the connections in the neural network in a manner that it minimizes a measure of the difference between the actual output vector of the network and the desired output vector. These adjustments in weights and their interactions, especially in hidden layers of the network, which are independent of input and the output, represent important meaningful features and regularities in the task domain of the network (Rumelhart et al., 1986). In mathematical words, an AE is a network that has the same number of input units as output units, and it is trained to generate an output x' that is close to the input x. The structure of this model may be viewed as the composition of two functions, f and g, that is $g \circ f$, where $f\colon \mathbb{R}^n \to \mathbb{R}^m$ is the encoder and where $g\colon \mathbb{R}^m \to \mathbb{R}^n$ is the decoder, respectively. Usually, $m \leq n$ because of the functionality of AE as a compressor of information or as a reductor of the high dimensionality of data. Thus, the input $x \in \mathbb{R}^n$ is mapped into a code $h = f(x)$, where $h \in \mathbb{R}^m$ represents meaningful features of x. Furthermore, $\mathbb{R}^m$ is defined as the latent space. On the other hand, the decoder g maps x into $g(f(x)) = g(h) = x'$, where $x' \approx x$. Therefore, $x' \in \mathbb{R}^n$ is defined as the reconstruction of x from h (Goodfellow et al., 2016).

To measure the degree of mismatch between input x and its reconstruction x', the weights w of the network are determined by minimizing an error function. In this work, we have used the average sum-of-square-errors, which has the form

$$E(w) = \frac{1}{N}\sum \parallel x'(x,w) - x \parallel^2 . \quad (1)$$

Equation (1) is called the loss function. $E(w)$ is the difference between x and x'. It is desirable to avoid having an AE as an identity mapping, which maps x' to x such that $x \approx x'$. Alternatively, one approach is to force the model to extract important and meaningful features from input data x. In this way, a useful model is a denoising autoencoder (DAE). If so, before training a DAE, input data x is altered with some kind of noise to give a modified input data $\tilde{x}$. In consequence, the output $x'(\tilde{x}, w)$ depends on $\tilde{x}$ and w (Vincent et al., 2008). Similarly, the degree of mismatch is given by the adjusted error function:

$$E(w) = \frac{1}{N}\sum \parallel x'(\tilde{x},w) - x \parallel^2 . \quad (2)$$

Notice from Equation (2) that x' depends on $\tilde{x}$ and w. Image denoising is considered an ill-posed problem. Owing to the difficulties in finding a unique solution, some techniques have been developed. A very simple and intuitive mathematical definition for noise on images is stated as follows:

$$\tilde{x} = x + n\,, \tag{3}$$

where $\tilde{x}$ is the distorted image, n is some additive noise added to x. Equation (3) represents the transformation of x into $\tilde{x}$, which are in the same Euclidean space. In a practical sense, denoising $\tilde{x}$ implies to process $\tilde{x}$ to obtain x', which is the best estimation of x. Such conceptualization is a starting point for understanding and dealing with structural modifications in images as a relevant fact among different study fields. According to Ilesanmi & Ilesanmi (2021), the most discussed types of noise are Gaussian noise, impulse noise, quantization noise, Poisson noise, speckle noise and salt-pepper noise. The presence of noise in images and its dealing with implies areas like medical imaging, remote sensing, military surveillance, biometrics and forensics, industrial, agricultural automation and many others. Denoising methods are classified as spatial domain methods, transform domain methods and CNN-based denoising methods. Spatial domain methods are divided into spatial domain filtering and variational denoising methods, which aim is to remove noise by calculating the gray value of each pixel based on the correlation between pixels/image patches in the original image. In contrast, transform domain methods include Fourier transform, cosine transform, wavelet domain, block-matching and 3D filtering. In general, with these methods, the characteristics of image information and noise are different in the transformation domain. CNN-based methods attempt to learn a mapping function by optimizing a loss function on a training set that contains distorted images (Fan et al., 2019). In a detailed way, CNN methods are described as more flexible and with more capacity respect to spatial and transform domain methods. CNN methods are divided into two approaches: the first one, for denoising general images; and the second one, for denoising specific images. In previous works, authors refer to general images which represent general purpose and for specific images that were created in specific fields like medicine, remote sensing, infrared sensing, etc. According to the authors, they concluded that different techniques related to CNN methods can remove all king of noise from images, and additionally, CNN architecture can be modified to remove bottleneck of vanish gradient (Ilesanmi & Ilesanmi, 2021). According to Venkataraman, (2022) and Zhang, (2018) two different models were used for denoising and compressing images. In both mentioned publications, a simple fully connected autoencoder (SAE) and a CAE were proposed as DAE. Both concluded that CAE had better performance than SAE. Inspired by these efforts, we have proposed CAE architecture with the aim of considering denoising images to generate novel images. Also, we focused on the performance of the CAE in the light of induced noise patterns.

In fact, according to consulted literature reviews on methods for image denoising using CNN architectures, it is possible to set up experimental work with simple CNN models, or even with implementing more advanced CNN models reported in the literature. Furthermore, there are different CNN architectures for different applications, e.g., classification (Navarro-Solís et al., 2024), denoising information, generative modeling, anomaly detection in medicine (López-Betancur et al., 2021), missing value imputation, image compression, dimensionality reduction, and so forth. So far, most of the ideas cited in the state of the art have proven useful.

3 Models and Methods

3.1 Dataset description and preprocessing

Initially, the CIFAR-10 dataset was downloaded from https://www.cs.toronto.edu/~kriz/cifar.htm. The CIFAR-10 dataset consists of 60,000 32x32 color images in 10 different classes, with 6000 images per class: airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck. The CIFAR-10 dataset is subdivide into two subsets: one with 50,000 for training a model, and a second one with 10,000 for validating it (*CIFAR-10 and CIFAR-100 datasets*, 2025). The dataset was saved in a CPU directory folder. Subsequently, each image was altered by randomly modifying pixel values in proportions of 30% (307 pixels), 60% (641 pixels), and 90% (912 pixels), respectively. The alteration process involved replacing proportions of pixels with completely black ([0, 0, 0]) or white ([255, 255, 255]) pixels. This process generated three distinct datasets, each containing 60,000 new images and subdivided similarly to the CIFAR-10 dataset. Fig. 1 presents one element from the validation subset of CIFAR-10 and its altered versions for validation, respectively.

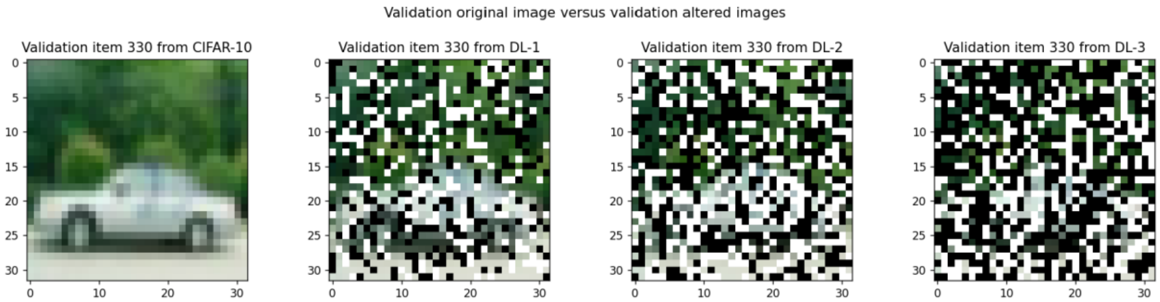


Fig. 1. Visual distortion in CIFAR-10: from 0% (original data from CIFAR-10) to 30%, 60% and 90% altered pixels.

3.2 Distortion levels on CIFAR-10 dataset

After the distortion process of CIFAR-10 dataset with three different levels, the resulting datasets were saved and labeled as follows: DL-1 for 30 % (307 of 1024 pixels), DL-2 for 30 % (614 of 1024 pixels) and DL-3 for 90% (912 of 1024 pixels).

3.3 The model architecture

Mostly, a CAE is structured in two basic parts: the first one, an encoder, which extracts features from input data and generates a representation into lower dimensions space (latent space); the second one, a decoder, that will take the latent representation to reconstruct the input data. In Fig. 2, a representation of the CAE architecture model is shown.

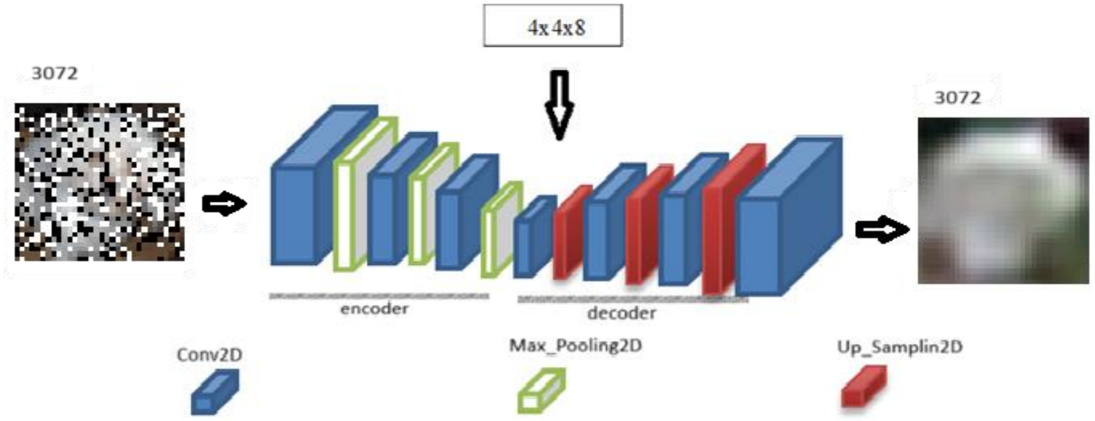


Fig. 2. CAE architecture model.

A detailed description is given below:

a) The encoder, which takes input color images of size 32x32x3 elements x of $\mathbb{R}^{3072}$, processes x through 4 convolutional layers and 3 max pooling layers, mapping them into a latent space of dimension 4x4x8 =128. Encoder maps x into an element of $\mathbb{R}^{128}$. Notice that after each convolutional layer there is a rectified linear layer (ReLU).

b) The decoder takes the code (latent value) and processes them through 3 convolutional layers and 3 transposed convolutional layers to reconstruct the original image. The resulting outputs are images with dimensions of 32x32x3.

The model was designed to generate new images with minimal alterations, aiming to match CIFAR-10 images as closely as possible. For that reason, the model was cloned and saved separately three times with different names; it guarantees that the weights of each version of the model are adjusted according to the distortion levels during training stage. Table 1. presents in more detail the type of layer and its output dimensions as the number of learnable parameters.

Table 1. CAE layer details.

Layer	Output	Learnable Parameters
conv2d (Conv2D)	[(None, 32, 32, 32)]	896
max_pooling2d (MaxPooling2D)	[(None, 16, 16, 32)]	0
conv2d (Conv2D)	[(None, 16, 16, 8)]	2312
max_pooling2d (MaxPooling2D)	[(None, 8, 8, 8)]	0
conv2d (Conv2D)	[(None, 4, 4, 8)]	584
max_pooling2d (MaxPooling2D)	[(None, 4, 4, 8)]	0
conv2d (Conv2D)	[(None, 4, 4, 8)]	584
Up_Sampling2D	[(None, 8, 8, 8)]	0
conv2d (Conv2D)	[(None, 8, 8, 32)]	2336
Up_Sampling2D	[(None, 16, 16, 32)]	0
conv2d (Conv2D)	[(None,16, 16, 32)]	9,248
Up_Sampling2D	[(None,32, 32, 32)]	0
conv2d (Conv2D)	[(None,32, 32, 3)]	867
Total of learnable parameters:		16827

3.4 Experiment setup

The model was programmed using Python version 3.11.7, TensorFlow and Keras libraries. Table 2. presents the characteristics of the computer resources implemented for this experiment.

Table 2. Specifications and setup.

Specifications	
Memory RAM	16 GB
Processor	13th Gen Intel(R) Core (TM) i7-13650HX 2.60 GHz
Graphics	NVIDIA GeForce RTX 4050 Laptop GPU
Operating Systems	Windows 11 Home Single Language

3.5 Training details

For the created datasets DL-1, DL-2, DL-3, the same hyperparameters values for each version model were set up for training stages. Each training phase took 500 epochs with the following hyperparameters values: learning-rate = 0.001, Adam optimizer parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 1 \times 10^{-8}$. The key optimizer parameters are set for default on Keras API. For the random initialization of weights was using a random seed equal to 42 (seed = 42). The Adam optimizer was implemented in this work following the descriptions provided by its creators (Kingma & Ba, 2017) and examples from https://keras.io.

4 Evaluation metrics

4.1 Performing Evaluation

Evaluating model performance was done in three steps: the first one, was through the estimation of loss function MSE which calculated the average of the squared difference errors between images from CIFAR-10 validation subsets and the outcome images after processing altered images from DL-1, DL-2, DL-3 validation subsets, respectively. The second step was through the implementation of three metrics: SSIM, PCC and PSNR which are useful for quality assessment of the new generated images, respect to CIFAR-10 images. Finally, the third step was through the application of the Shannon metric which measures the randomness of pixel values.

4.1.1 Structural Similarity Index Metric

The SSIM is an algorithm that captures from two images x and y three aspects: luminance $l(x,y)$, contrast $c(x,y)$ and structure $s(x,y)$. These three aspects are combined as follows:

$$SSIM(x,y) = l(x,y)c(x,y)s(x,y) = \frac{(2\mu_x\mu_y+C_1)(2\sigma_{xy}+C_2)}{({\mu_x}^2+{\mu_y}^2+C_1)({\sigma_x}^2+{\sigma_y}^2+C_2)}, \qquad (4)$$

where x and y are discrete signals, μ_x and μ_y are the respective mean intensity. σ_x and σ_y are estimations of the signals contrast, respectively. σ_{xy} is the correlation coefficient between x and y. The constants C_1 and C_2 in Equation (4) are included to avoid instability when ${\mu_x}^2 + {\mu_y}^2$ and ${\sigma_x}^2 + {\sigma_y}^2$ are close to cero. SSIM index ranges from -1 to 1. If the score is close to 1, that indicates that similarly is almost total. If it is close to 0, then similarity is low. And for negative values there is not any structural similarity (Wang et al., 2004; Nilsson & Akenine-Möller, 2020). In this generation process of images, the outcomes from CAE and the corresponding ones from CIFAR-10 must be compared one-to-one to evaluate the model performance, but basically the structural similarity of them, i.e., the quality of the outcomes.

4.1.2 Pearson Correlation Coefficient

Similarly, in this work the PCC (r) measures the linear relationship between two random vectors (images) x and y, ranging from -1 to 1. r is defined as follows:

$$r = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(x_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}}, \qquad (5)$$

where $\bar{x}$ and $\bar{y}$ are the means of x and y, and n is the length of the vectors. The coefficient r is interpreted as follows:

- If $r = 1$, there is a perfect linear correlation. As one variable increases, the other increases proportionally.
- If $0 < r < 1$, there is a positive correlation. A higher r indicates higher linear correlation.
- If $r = 0$, there is no linear correlation.
- If $-1 < r < 0$, negative linear correlation. A lower r indicates a stronger negative relationship.
- If $r = -1$, there is a perfect negative linear correlation. As one variable increases, the other decreases proportionally.

In this work PCC is applied over all pixel images. Also, this metric is known as Correlation Criteria (CC) or Dependency Measure (DM) that is based on relevance (predictive power) of feature image. The predictive power is computed by finding the correlation between features of two variables (Ghojogh et al., 2019).

4.1.3 Peak Signal-to-Noise Ratio

As third metric option for quality is PSNR. Given a ground truth image x, the PSNR of a generated image x' is defined by

$$PSNR(x,x') = 10 * log_{10}(\frac{255^2}{\|x-x'\|_2^2}), \qquad (6)$$

where 255 is maximum pixel value in images. In the preprocessing stage of the CIFAR-10 dataset, all images were normalized from 0 to 1. For that reason, the Equation (6) takes the form:

$$PSNR(x,x') = 10 * log_{10}(\frac{1}{\|x-x'\|_2^2}). \qquad (7)$$

The PSNR metric values are given in decibels (dB).

- From 40 to higher, the new image is indistinguishable from the original.
- From 30 to 40, the new image is a very good quality image.
- From 20 to 30, the new image has an acceptable quality, with visible distortions.
- Below 20 the new image is of very poor quality, with significant distortions.

To see more about these metric and its applications refer to other works (Fan et al., 2019; Ilesanmi & Ilesanmi, 2021; Nilsson & Akenine-Möller, 2020).

4.1.4 The Shannon metric

It is important to have insights into the complexity, randomness, and informational content of data. In particular, randomness of pixel values in a generated image is compared with that in the original one. A low entropy indicates low variability. A high entropy means that an image has uniform pixel values, and it suggests high variability for noise or texture. The Shannon entropy H is defined as follows:

$$H(x) = -\sum_{i=1}^{n} p(x_i) ln_2 p(x_i) \, , \tag{8}$$

where x is random image, x_i is a pixel value of x, $p(x_i)$ is the probability associated with x_i and $p(x_i)ln_2p(x_i)$ is entropy in bits. Also, this metric is known as Mutual Information (MI) or Information Gain (IG) and is the measure of dependence or shared information between two random variables (Ghojogh et al., 2019).

5 Results and Discussions

In this work, we confirm that CNN-based models have demonstrated potential for generating information. In some works, Venkataraman, (2022) and Zhang, (2018), proposals of CAE and a full-connected autoencoder were compared on denoising images. CAE was better than fully connected. According to Venkataraman, (2022) in 20 epochs for training, the CAE reached a validation loss of 0.0871, and the fully connected autoencoder reached a validation loss of 0.2305 in a training time of 100 seconds. In this sense, for this work, MSE loss function (left graph) over 500 epochs tended to decline slowly for higher distortion levels. This fact confirms that higher distortion levels make it more difficult to adjust model weights.

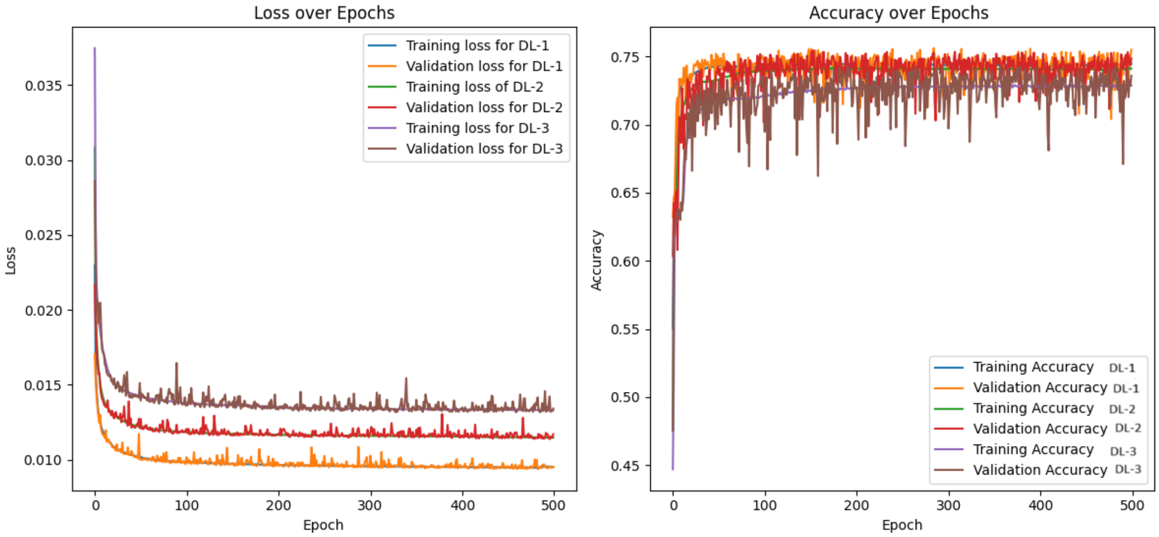


Fig. 3. MSE and accuracy metrics were applied over 500 epochs for DL-1, DL-2 and DL-3.

Table 3. presents the model performance trained independently on the three datasets: DL-1, DL-2 and DL-3. According to the first row, MSE function declined more slowly during training and validation stages for DL-2 and DL-3 than for DL-1.

Similarly, the second row presents challenging improvements for accuracy metric with validation images from DL-3. In contrast, for DL-1 and DL-2 it appears to reach a higher value over 500 epochs. With respect to time for training and validation, it increases with higher levels of distortion. The training and validations times, in seconds, for each model amounted as follows: 6113.49 and 1.48 for DL-1, 6826.38 and 1.58 for DL-2, and finally, 7890.34 and 1.60 for DL-3.

Table 3. Validation loss and Accuracy.

CAE metrics	DL-1	DL-2	DL-3
Validation Loss	0.0095	0.0117	0.0134
Validation Accuracy	0.7549	0.7486	0.7359

Table 4. Experimental results of SSIM, PCC, and PSNR on four randomly generated images after denoising, with their versions from the validation subsets of DL-1, DL-2, and DL-3, compared with the version from the CIFAR-10 validation subset.

Image *	**SSIM**			**PCC**			**PSNR in dB**		
330	0.6830	0.6351	0.6044	0.9235	0.9073	0.8966	20.08	19.25	18.77
8551	0.6146	0.5545	0.6044	0.9358	0.9168	0.9005	20.54	19.49	18.63
5555	0.4148	0.3939	0.3251	0.6259	0.5790	0.5034	24.62	23.94	23.71
70	0.6753	0.5869	0.4697	0.8998	0.8597	0.8109	26.04	24.67	23.77

*Four images (item numbers in the first column) were randomly selected from validation subset from CIFAR-10. The metrics were applied to compare them with the three generated images (middle columns for each metric) resulting from applying the CEA to the altered images from the validation subsets from DL-1, DL-2 and DL-3, respectively.

5.1 Quality assessment for generated images: performance insights

To assess objectively the quality for generated images by the CAE respect to those from CIFAR-10, three metrics were applied: SSIM, PCC and PSNR. Table 4. presents the results of applying these metrics to 330th, 8551st, 5555th, and 70th images, which were randomly selected from the validation subsets of the CIFAR-10, DL-1, DL-2, and DL-3, respectively. Because of 10,000 images in validation subsets from CIFAR-10, refer to Fig. 4. to see image 8551 (a ship) with its altered versions.

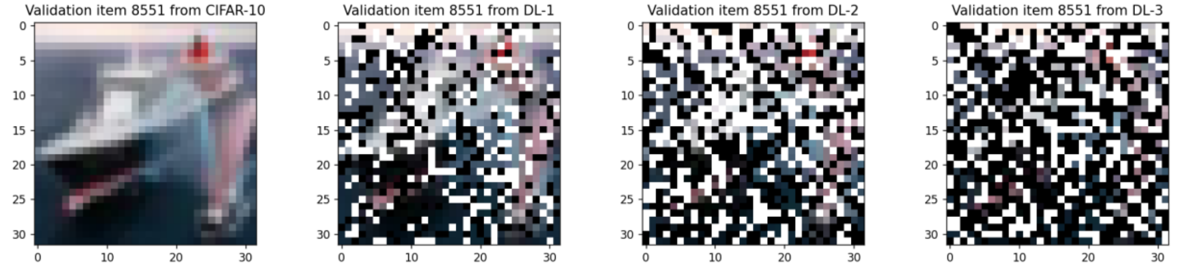

Validation item 8551 from CIFAR-10
Validation item 8551 from DL-1
Validation item 8551 from DL-2
Validation item 8551 from DL-3
0
5
10
15
20
25
30
0
10
20
30

Fig. 4. Items 8551 randomly selected from validation subsets: CIFAR-10, DL1-, DL-2 and DL-3, respectively.

In Table 4, SSIM index score describes moderate structural similarity between 330, 8551 and 70 compared to the generated images. In contrast, item 5555 has significant structural differences with the CAE outcomes. Refer to Fig. 5. to have a visual comparison between item 5555 and the outcomes of the model.

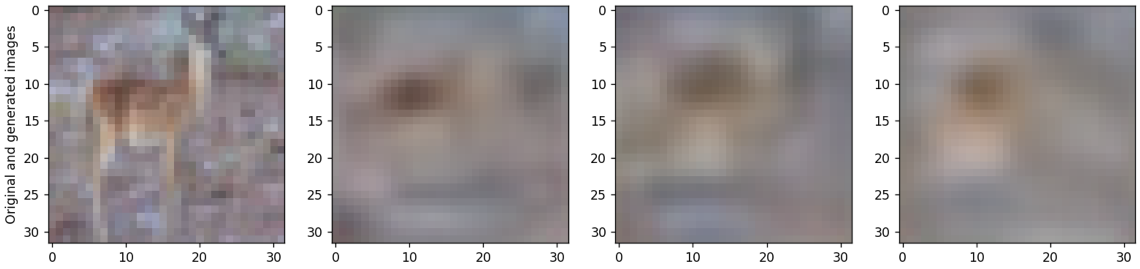


Fig. 5. Item 5555 from CIFAR-10 and the model outputs.

According to the PCC, for items 330, 8551 and 70 measured a strength linear correlation with the CAE outcomes. So, notice that r is close to 1. In contrast, PCC for item 5555 indicates a weak linear correlation. Finally, PSNR indicates respect to generated outcomes, particularly about the 330 and 8551 items, a more degradation or less quality. In contrast, with respect to items 5555 and 70, PSNR shows more quality in the generated images. Refer to Fig. 6. to have a visual comparison between item 8551 and the generated ones.

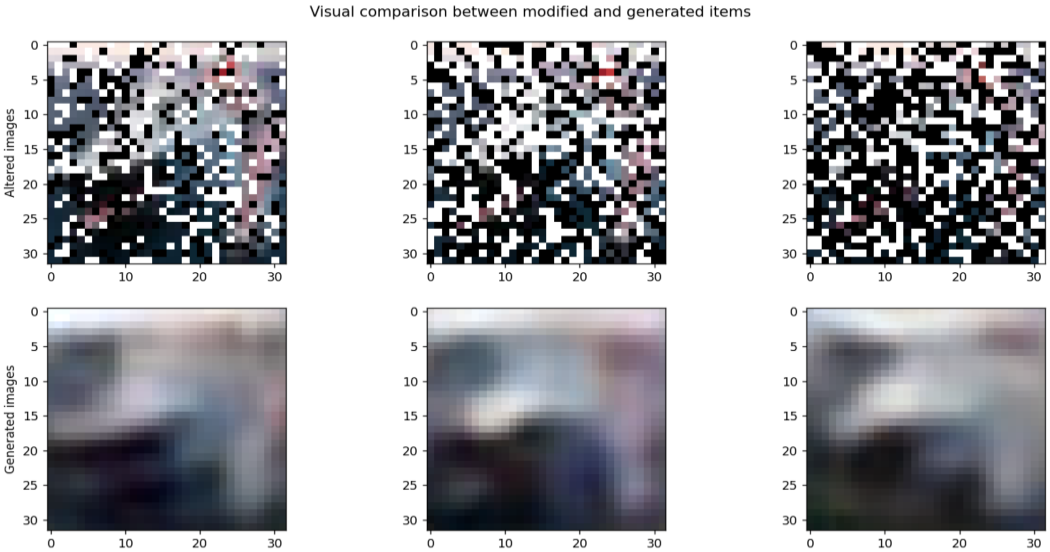


Fig. 6. Images generated (second row) with CAE using items no. 8551 from validation datasets DL-1, DL-2 and DL-3.

SSIM index as a popular measure in many different scientific projects for almost two decades (Nilsson & Akenine-Möller, 2020). However, in Старовойтов (2019) authors demonstrated that SSIM index or any linear transformation of it is not a metric in the mathematical sense. Additionally, in this work, it was discussed that SSIM cannot correctly determine the similarity between two images, just only similarity of visually close images of the same scene. Finally, authors concluded that PCC is a more accurate measure of similarity and dissimilarity of the compared images than the SSIM. As a factual element for discussion about using SSIM, in Fan et al. (2019) implemented SSIM and PSNR metrics to evaluate the denoising results on Lena image that was corrupted with Gaussian noise with $\sigma = 30$ and filtered with the methods: Weiner filtering (PSNR = 27.81 dB; SSIM= 0.707); Bilateral filtering (PSNR = 27.88 dB; SSIM= 0.712); PCA method (PSNR = 26.68 dB; SSIM= 0.596); Wavelet transform domain method (PSNR = 21.74 dB; SSIM= 0.316); Collaborative filtering: BM3D (PSNR = 31.26 dB; SSIM= 0.845). Hint: dB for decibels. The previous work, based on the PSNR and PCC values, concluded that 3D Filtering (BM3D) method is better than the other ones, and with a big potential for noise reduction and edge protection. In this sense, the assessment results in Table 4. obtained in this work, SSIM and PCC values are lower for image 5555, but with a larger value for PSNR. Particularly, that indicates less structural similarities, but with a higher quality. It is noticeable about metrics values, that for some generated images, there is not a linear correlation. However, to have general conclusions about the model, it is necessary to estimate ranges of values based on the Table 4: SSIM [0.3251, 0.6830], PCC [0.5034, 0.9358], and PSNR [18.63 dB, 26.04 dB], respectively.

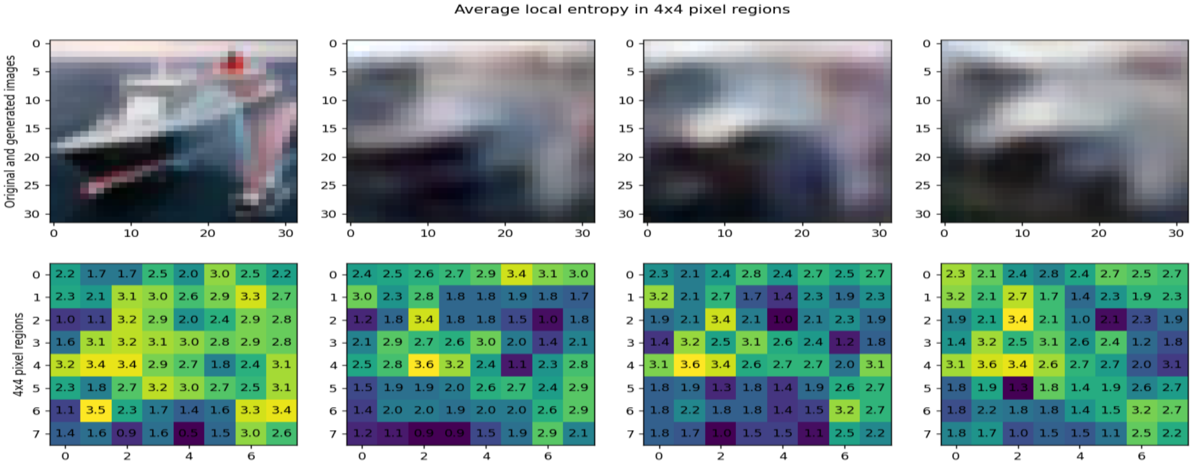
5.2 Performance: randomness assessment with Shannon entropy

The randomness assessment of pixel values is another important aspect of the generated images with CAE. It is important to know that this assessment of entropy is applied just to one individual image. In this work we calculated separately the LAE by local regions of 4x4 pixels, TE of each selected randomly image. According to this metric, a higher value indicates higher uncertainty on information. In this sense, entropy is an inherent property of pixel values, and it is not dependent on external factors. Nevertheless, in this experimental work, the entropy values are very similar to each other, which could suggest a potential relationship between them. The resulting values ranged as follows: [1.25 bits, 2.49 bits] for local regions, and [6.61 bits, 9.71 bits] for the total image, respectively. Refer to Table 5.

Table 5. LAE of 64 local 4x4 pixel regions and the TE of the entire validation image from CIFAR-10 and the corresponding generated ones.

Validation images versus generated ones.	**Local Average Entropy**	**Total Entropy Image**
Image 330	2.49	9.12
Image generated from DL-1	2.18	9.04
Image generated from DL-2	2.17	8.92
Image generated from DL-3	2.03	8.75
Image 8551	2.43	9.71
Image generated from DL-1	2.19	9.40
Image generated from DL-2	2.19	9.61
Image generated from DL-3	2.22	9.68
Image 5555	2.28	7.79
Image generated from DL-1	1.44	7.86
Image generated from DL-2	1.43	8.36
Image generated from DL-3	1.25	7.91
Image 70	1.69	6.61
Image generated from DL-1	1.51	7.44
Image generated from DL-2	1.44	7.61
Image generated from DL-3	1.31	7.55

In Table 5 LAE in most cases is higher for validation images than that from generated images, and that indicates a higher uncertainty of pixel values. The validations images are in their original condition. In this sense, generated images have less randomness. In contrast, TE shows a little lower value for images of 70 and 5555 for validation images than that for generated images.



	0		2		4		6	
0	2.2	1.7	1.7	2.5	2.0	3.0	2.5	2.2
1	2.3	2.1	3.1	3.0	2.6	2.9	3.3	2.7
2	1.0	1.1	3.2	2.9	2.0	2.4	2.9	2.8
3	1.6	3.1	3.2	3.1	3.0	2.8	2.9	2.8
4	3.2	3.4	3.4	2.9	2.7	1.8	2.4	3.1
5	2.3	1.8	2.7	3.2	3.0	2.7	2.5	3.1
6	1.1	3.5	2.3	1.7	1.4	1.6	3.3	3.4
7	1.4	1.6	0.9	1.6	0.5	1.5	3.0	2.6

	0		2		4		6	
0	2.4	2.5	2.6	2.7	2.9	3.4	3.1	3.0
1	3.0	2.3	2.8	1.8	1.8	1.9	1.8	1.7
2	1.2	1.8	3.4	1.8	1.8	1.5	1.0	1.8
3	2.1	2.9	2.7	2.6	3.0	2.0	1.4	2.1
4	2.5	2.8	3.6	3.2	2.4	1.1	2.3	2.8
5	1.5	1.9	1.9	2.0	2.6	2.7	2.4	2.9
6	1.4	2.0	2.0	1.9	2.0	2.0	2.6	2.9
7	1.2	1.1	0.9	0.9	1.5	1.9	2.9	2.1

	0		2		4		6	
0	2.3	2.1	2.4	2.8	2.4	2.7	2.5	2.7
1	3.2	2.1	2.7	1.7	1.4	2.3	1.9	2.3
2	1.9	2.1	3.4	2.1	1.0	2.1	2.3	1.9
3	1.4	3.2	2.5	3.1	2.6	2.4	1.2	1.8
4	3.1	3.6	3.4	2.6	2.7	2.7	2.0	3.1
5	1.8	1.9	1.3	1.8	1.4	1.9	2.6	2.7
6	1.8	2.2	1.8	1.8	1.4	1.5	3.2	2.7
7	1.8	1.7	1.0	1.5	1.5	1.1	2.5	2.2

	0		2		4		6	
0	2.3	2.1	2.4	2.8	2.4	2.7	2.5	2.7
1	3.2	2.1	2.7	1.7	1.4	2.3	1.9	2.3
2	1.9	2.1	3.4	2.1	1.0	2.1	2.3	1.9
3	1.4	3.2	2.5	3.1	2.6	2.4	1.2	1.8
4	3.1	3.6	3.4	2.6	2.7	2.7	2.0	3.1
5	1.8	1.9	1.3	1.8	1.4	1.9	2.6	2.7
6	1.8	2.2	1.8	1.8	1.4	1.5	3.2	2.7
7	1.8	1.7	1.0	1.5	1.5	1.1	2.5	2.2

Fig. 7. The LAE map is shown for image 8551 from the validation subset of CIFAR-10 and for the corresponding generated images.

Fig. 7. presents the distribution of local average entropy (second row) in 4x4 pixel regions. For yellow color regions LE values are higher and gradually decreased values for the other colors. Also, Fig. 7 presents a visual patterns of the colors according with higher and lower intensity.

5.3 Conclusions

According to the results obtained in this experiment, we have concluded that:

- Despite its simplicity, the CAE effectively learns meaningful features from corrupted images with a relatively small number of parameters. The ratio of trainable parameters to validation images highlights its efficiency; 16827 learnable parameters to 10,000 validation images.
- The study avoids relying on subjective human evaluation and instead emphasizes objective metrics to analyze image quality and structural information.

- The study measures image similarity using SSIM and PCC, showing moderate structural resemblance and strong linear correlation between generated and original images.
- Lower Shannon entropy in generated images suggests that the CAE may not fully preserve the original dataset's complexity, indicating a need for architectural adjustments

5.4 Future Work

The CAE model coded 60,000 vectors from a 3072-dimensional to a 128-dimensional vector space. Future work could investigate the characteristics of 128-dimensional latent space. Furthermore, it is a high dimensionality reduction problem. Additionally, the datasets generated for altering the pixel values and the outputs images after feeding the CAE give a suitable opportunity for applying stochastic embedding methods to understand about feature extraction and data visualization: Stochastic Neighbor Embedding (SNE), Student's t-Distributed Stochastic Method (t-SNE) and Uniform Manifold Approximation Projection (UMAP).

Acknowledgements
This project is supported by CONAHCYT and the Instituto Politécnico Nacional de México.

References

Старовойтов, В. В. (2019). Индекс SSIM не является метрикой и плохо оценивает сходство изображений. *Системный анализ и прикладная информатика*, 2, 12–17. https://doi.org/10.21122/2309-4923-2019-2-12-17

Fan, L., Zhang, F., Fan, H., & Zhang, C. (2019). Brief review of image denoising techniques. *Visual Computing for Industry, Biomedicine, and Art, 2*(1), 7. https://doi.org/10.1186/s42492-019-0016-7

Ghojogh, B., et al. (2019). Feature selection and feature extraction in pattern analysis: A literature review. *arXiv*. https://doi.org/10.48550/arXiv.1905.02845

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Ilesanmi, A. E., & Ilesanmi, T. O. (2021). Methods for image denoising using convolutional neural network: A review. *Complex & Intelligent Systems, 7*(5), 2179–2198. https://doi.org/10.1007/s40747-021-00428-4

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv*. https://doi.org/10.48550/arXiv.1412.6980

Krizhevsky, A. (2009). *Learning multiple layers of features from tiny images (CIFAR-10 and CIFAR-100 datasets)*. https://www.cs.toronto.edu/~kriz/cifar.html

López-Betancur, D., Bosco Durán, R., Guerrero-Méndez, C., Zambrano Rodríguez, R., & Saucedo Anaya, T. (2021). Comparación de arquitecturas… *Computación y Sistemas, 25*(3), 601–615. https://doi.org/10.13053/cys-25-3-3453

López-Betancur, D., González-Ramírez, E., Guerrero-Méndez, C., Saucedo-Anaya, T., Rivera, M. M., Olmos-Trujillo, E., & Gómez-Jiménez, S. (2024). Evaluation of optimization algorithms for measurement of suspended solids. *Water, 16*(13), Article 1761. https://doi.org/10.3390/w16131761

Navarro-Solís, D., Guerrero-Méndez, C., Saucedo-Anaya, T., López-Betancur, D., Silva, L., Robles-Guerrero, A., & Gómez-Jiménez, S. (2024). Analysis of convolutional neural network models for classifying the quality of dried chili peppers (*Capsicum annuum* L.). En H. Calvo et al. (Eds.), *Advances in Computational Intelligence: MICAI 2023 International Workshops* (pp. 116–131). Springer. https://doi.org/10.1007/978-3-031-51940-6_10

Nilsson, J., & Akenine-Möller, T. (2020). Understanding SSIM. *arXiv*. http://arxiv.org/abs/2006.13846

Pappas, T. N., Safranek, R. J., & Chen, J. (2000). Perceptual criteria for image quality evaluation. In *Handbook of image and video processing*.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature, 323*(6088), 533–536. https://doi.org/10.1038/323533a0

Sparavigna, A. C. (2019). Entropy in image analysis. *Entropy, 21*(5), 502. https://doi.org/10.3390/e21050502

Venkataraman, P. (2022). Image denoising using convolutional autoencoder. *arXiv*. https://doi.org/10.48550/arXiv.2207.11771

Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. En *Proceedings of the 25th International Conference on Machine Learning (ICML)* (pp. 1096–1103). https://doi.org/10.1145/1390156.1390294

Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing, 13*(4), 600–612. https://doi.org/10.1109/TIP.2003.819861

Zhang, Y. (2018). *A better autoencoder for image: Convolutional autoencoder* [Manuscrito no publicado]. http://users.cecs.anu.edu.au/Tom.Gedeon/conf/ABCs2018/paper/ABCs2018_paper_58.pdf