Identification and Improvement of Image Similarity using Autoencoder

Suresh Merugu
School of Computer Science, University of Southampton Malaysia, Malaysia
s.merugu@soton.ac.uk (corresponding author)

Rajesh Yadav
School of Computer Science, University of Southampton Malaysia, Malaysia
rajeshyadav@soton.ac.uk

Venkatesh Pathi
Department of Computer Science and Engineering, CMR College of Engineering & Technology, India
venkatesh.pathi6@gmail.com

Herbert Raj Perianayagam
School of Computer Science, University of Southampton Malaysia, Malaysia
herbert.r.perianayagam@soton.ac.uk

Received: 19 April 2024 | Revised: 6 May 2024 | Accepted: 7 May 2024

 DOI: https://doi.org/10.48084/etasr.7548

ABSTRACT

Identifying the similarity between fine-grained images requires sophisticated techniques. This study presents a deep learning approach to the image similarity problem as an unsupervised learning task. The proposed autoencoder, built on a Deep Neural Network (DNN), autonomously learns image representations by computing cosine similarity distances between extracted features. This paper presents several applications, including training the autoencoder, transforming images, and evaluating the DNN model. In each instance, the generated images exhibit sharpness and closely resemble natural photographs, demonstrating the effectiveness and versatility of the proposed deep learning framework in computer vision tasks. The results suggest that the proposed approach is well-suited for tasks that require accurate image similarity assessments and image generation, highlighting its potential for various applications in image retrieval, data augmentation, and pattern recognition. This study contributes to the advancement of the computer vision field by providing a robust and efficient method for learning image representations and evaluating image similarity in an unsupervised manner.

Keywords-Deep Neural Network (DNN); autoencoder; unsupervised learning; image similarity; cosine similarity

I. INTRODUCTION

In recent years, the field of neural network-based image generation has seen a remarkable surge in interest and research activity. This increased interest is driven by the diverse and transformative applications of image generation techniques across various domains, including unsupervised and semi-supervised learning, generative modeling, representation learning, synthesis evaluation, and even extending to 3D representation learning and video prediction. Central to these applications is the ability of neural networks to learn and generate complex visual data, ranging from realistic images to abstract representations. Among the fundamental challenges in this domain is the task of reconstructing images from learned feature representations. The fidelity and quality of the reconstructed images are critical for the effectiveness and practical applicability of image generation methods. However, traditional approaches often struggle to preserve fine-grained details, resulting in reconstructed images that are blurry or lack important visual attributes.

II. BACKGROUND STUDY

This study aimed to develop an autoencoder model capable of autonomously learning from data and performing accurate image reconstructions that closely resemble the original inputs. Autoencoders represent a class of neural network architectures consisting of an encoder and a decoder. The encoder compresses the input data into a latent representation, capturing essential features and patterns, while the decoder reconstructs the original input from this compressed representation. This

framework allows for the unsupervised learning of feature representations that effectively capture the intrinsic characteristics of the input data.

To identify and measure similarity between images based on their learned representations, the extracted latent features are subjected to a cosine similarity function, often in conjunction with Principal Component Analysis (PCA). The cosine similarity metric measures the angle between feature vectors and is particularly effective in quantifying similarity in high-dimensional spaces. PCA, on the other hand, reduces the dimensionality of feature vectors while retaining the most significant variance, facilitating efficient and robust similarity comparisons. Convolutional Neural Networks (CNNs) play a central role in this study by providing powerful and effective feature representations [1, 2]. CNNs excel at extracting hierarchical and spatially invariant features from images, making them well-suited for tasks such as image reconstruction and similarity analysis. These networks are capable of learning representations that are robust to minor distortions but sensitive to perceptually significant image attributes, such as edges, textures, and object shapes. Training a decoder network to reverse the encoding process is essential to ensure that the reconstructed images closely resemble the original inputs.

In summary, this study aims to leverage advanced neural network architectures, particularly autoencoder DNN, to address the challenge of image reconstruction and similarity analysis. By harnessing the power of deep learning and sophisticated feature representations, the goal is to develop robust and effective methods for generating high-quality images and quantifying their visual similarity. The insights gained from this research have broad implications for various applications that require image generation, representation learning, and similarity analysis in complex visual datasets.

III. LITERATURE REVIEW

In [3], a deep Siamese network, called SimNet, was trained on pairs of effective and bad pictures using a unique online pair-mining strategy. A multi-scale CNN was also proposed, demonstrating the superiority of multi-scale Siamese networks in capturing fine-grained picture similarities over conventional CNNs, combining top- and lower-layer embeddings. In [4], fine-grained image similarity learning was achieved using deep-ranking methods, demonstrating superior classification performance. In [5], sparse online learning techniques were proposed to enhance image similarity efficiency and scalability. In [6], DeePSiM was proposed to generate high-resolution images from compressed abstract representations. In [7], a loss function was applied to a variational autoencoder for optical character recognition and Quranic image similarity matching. In [8], bit-scalable deep hashing was combined with regularized similarity learning to facilitate efficient image retrieval and person reidentification. In [9], a visual imitation learning framework was proposed, using a convolutional autoencoder, to enable robotic learning actions based on sample videos and actions. In [10], an SVM approach was proposed for image classification. In [11], a binary object detection model was proposed to assist visually impaired people. In [12], Weber's law-based regularization was used to de-blur images, enhancing image quality restoration and sharpness for visually impaired users.

In [13], a novel method was proposed to enhance the similarity assessment of food images. In [14], an improved triplet network was used with spatial pyramid pooling, showing improved performance in image similarity assessment. In [15], CNNs were used for sub-scene target detection. In [16], a CNN was proposed to capture visual similarity. Leveraging inter-image similarity and an ensemble of extreme learners for fixation prediction significantly improves accuracy within deep learning frameworks. In [17], an ensemble of extreme learning machines was used to measure the saliency of input images. In [18], a method was proposed to highlight the regions of images that contribute more to pairwise image similarity. This approach was extended by implementing various pooling techniques, enabling image similarity assessments on objects or sub-regions within the query image. In [19], the SOLIS scheme was proposed to optimize image similarity with sparse and high-dimensional image representations. However, most existing methods have the following limitations:

- Metrics and comparisons are limited to specific datasets and architectures, reducing generalizability to broader image similarity tasks.
- Models effective in controlled settings may struggle with real-world deployment due to interpretability, computational efficiency, and usability issues.

IV. DATASET USED

To train an unsupervised machine learning model, a dataset consisting of images without any associated labels is needed. This study collected images from various sources. Web scraping techniques were used to collect animal images, such as foxes, tigers, and wolves, from Google. These images were obtained from publicly available sources on the Internet. Additionally, the MNIST dataset was incorporated, which is freely accessible through the Keras library. The dataset included a total of 4738 images, combining those obtained through web scraping and the MNIST dataset. These images were used for both training and testing the proposed model. It is important to note that the MNIST dataset is widely used in the machine learning community and is readily available for experimentation purposes due to its open nature.

V. METHODOLOGY

A. Autoencoder

An autoencoder is a kind of ANN that is used to find efficient encodings of information in an uncontrolled way. The purpose of an autoencoder is to look for the representation (encoding) of a series of facts, such as images. The autoencoder structure has an encoder and a decoder part in its architecture, which may be described as transitions φ and ψ, that is:

$$\emptyset: X \rightarrow F \quad (1)$$

$$\emptyset: F \rightarrow X \quad (2)$$

$$\varphi, \psi = \arg\min \| X - (\psi . \varphi) X \| \quad (3)$$

In the handiest case, given one hidden layer, the encoder part takes the entered $x \in Rd = X$ and maps it to $h \in Rp = F$:

$$h = \sigma(Wx + b) \quad (4)$$

The picture h is typically called code, hidden variables, or hidden view. Here, σ is the element-sensible activation characteristic including a sigmoid characteristic or a rectified linear unit, W is a weight matrix and b is an offset vector. Weights and pre-loads are usually randomly initialized, after which update iteratively at some point of learning via backpropagation. After that, the decoder layer converts h into a reconstruction of x' with the identical form a x:

$$x' = \sigma'(W'h + b') \quad (5)$$

where σ', W', and b' for the decoder can be mapped to the corresponding σ, W, and b for the encoder. Autoencoders are trained to reduce recovery errors, such as root mean square error, often referred to as losses:

$$L(x,x') = \left||x - x'|\right|^2$$

$$= \left||x - \sigma'\big(W\big(\sigma(Wx + b)\big) + b\big)|\right|^2 \quad (6)$$

where x is normally averaged over a few input training sets. The autoencoder is trained via backpropagation of the error, much like a normal feedforward neural network. In the perfect setting, one must be capable of customizing the code measurement and the model ability on the idea difficulties in the distribution of the records to be modeled. One way to achieve this is to make the most of the model variations called regularized autoencoders. Table I and Figure 2 detail the architecture of the proposed autoencoder. Its architecture consists of an input layer, hidden layers, and an output layer. The input layer receives the input data, which are then encoded in the hidden layers to a latent representation. The latent representation is then decoded back to the output layer, aiming to reconstruct the input data. Each layer consists of a specific number of nodes or neurons that contribute to the encoding or decoding processes.

TABLE I. AUTOENCODER ARCHITECTURE

Layer (type)	Output Shape	Param
#input1 (InputLayer)	(None, 128, 128, 3)	0
conv2d (Conv2D)	(None, 128, 128, 512)	14336
maxpooling2d (MaxPooling2D)	(None, 64, 64, 512)	0
conv2d1 (Conv2D)	(None, 64, 64, 512)	2359808
maxpooling2d1 (MaxPooling2)	(None, 32, 32, 512)	0
conv2d-2 (Conv2D)	(None, 32, 32, 128)	589952
conv2d-3 (Conv2D)	(None, 32, 32, 128)	147584
maxpooling2d-2 (MaxPooling2)	(None, 16, 16, 128)	0
conv2d-4 (Conv2D)	(None, 16, 16, 128)	147584
maxpooling2d-3 (MaxPooling2)	(None, 8, 8, 128)	0
conv2d-5 (Conv2D)	(None, 8, 8, 128)	147584
up-sampling2d (UpSampling2D)	(None, 16, 16, 128)	0
conv2d-6 (Conv2D)	(None, 16, 16, 128)	147584
conv2d-7 (Conv2D)	(None, 16, 16, 128)	147584
up-sampling2d-1 (UpSampling2)	(None, 32, 32, 128)	0
conv2d-8 (Conv2D)	(None, 32, 32, 512)	590336
up-sampling2d-2 (UpSampling2)	(None, 64, 64, 512)	0
conv2d-9 (Conv2D)	(None, 64, 64, 512)	2359808
up-sampling2d-3 (UpSampling2)	(None, 128, 128, 512)	0
conv2d-10 (Conv2D)	(None, 128, 128, 3)	13827

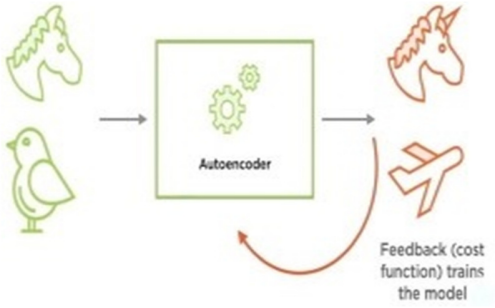

Fig. 1. Basic understanding of an autoencoder.

Fig. 2. Proposed autoencoder architecture.

B. Overfitting

Dropout regularization, early stopping, batch normalization, and data augmentation were applied to prevent overfitting.

Dropout regularization randomly disables a portion of the neurons during training to enhance generalization. Early stopping stops training when the validation error stops decreasing, minimizing overfitting. Batch normalization normalizes the input of each layer to stabilize and accelerate training. Increasing pictures with random transformations (no picture is visible twice) improves the results of the autoencoder and prevents overfitting. Each projection and picture increases the tendency to break random correlations. Dropout is largely equal to the L1 norm.

C. Cosine Similarity

Cosine similarity was used as a similarity metric for comparing vectors at the latent space (encoded representation). In this way, the model can measure the similarity between different input data points based on direction rather than magnitude, providing insights into their closeness within the learned feature space. Cosine similarity is a degree of similarity among non-zero vectors of internal product space and identifies the cosine of the perspective among them.

$$cosineDist = 1 - cosinesimilarity \quad (7)$$

$$cosinesimilarity = \cos(\theta) \quad (8)$$

where cos(θ) is the angle between item1 and item2, and cosine similarity ranges between [-1, 1], as shown in Figure 5.

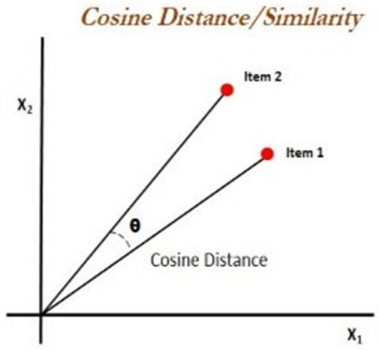

Fig. 3. Cosine similarity/cosine distance.

VI. PERFORMANCE EVALUATION

The proposed autoencoder exhibited extended training periods but demonstrated exceptional prediction efficiency. Google Colab facilitates the utilization of Colab GPU and TPU, delivering computational capabilities that enable image processing at around 55 ms (TP90). Although processing images and training models may take varying durations, optimizing batch size, fine-tuning hyperparameters, and exploring modifications in network layers proved pivotal in enhancing model performance, contingent upon the dataset for further refinement. Using an autoencoder to perform picture similarity tasks involving a reference image showed superior performance in image similarity tasks compared to traditional CNNs. The implementation of cosine similarity for image comparison surpasses the conventional Euclidean distance approach, offering a more sophisticated method for determining image similarities.

VII. RESULTS AND DISCUSSION

Figure 4-7 represent key aspects of the proposed model evaluation process. This study meticulously assessed the performance of the proposed autoencoder model on test data, rigorously experimenting with various hyperparameters and tuning techniques to optimize its effectiveness. It is crucial to emphasize that the success of these optimizations heavily relies on the characteristics and nuances of the dataset itself. The autoencoder was trained using backpropagation of the error, much like a normal feedforward neural network *F*. Figure 4 presents the results of the initial experimentation of the proposed model with baseline hyperparameters. Figure 5 illustrates the impact of adjusting specific hyperparameters, showcasing how variations in these settings influence the model's results.

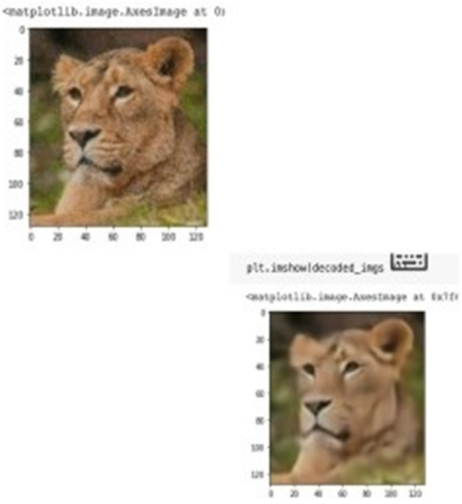

Fig. 4. Validations results of the autoencoder.

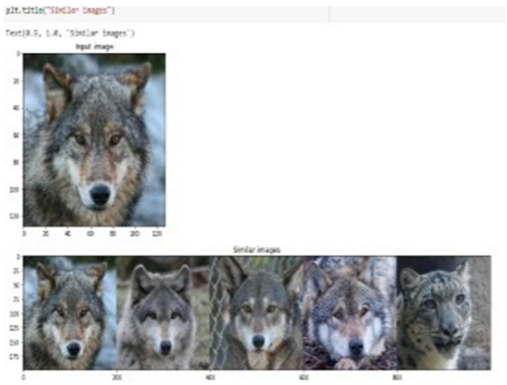

Fig. 5. Results with hyperparameter tuning (1st image).

Figures 6, 7, and 8 delve deeper into the effects of fine-tuning certain parameters, demonstrating how nuanced adjustments can lead to noticeable improvements or refinements in the model's performance. These results provide a comprehensive overview of the proposed model findings, synthesizing the results of proposed model experimentation efforts and offering valuable insights for future optimization endeavors. Looking ahead, there is considerable potential to

incorporate more advanced deep-learning architectures into future research.

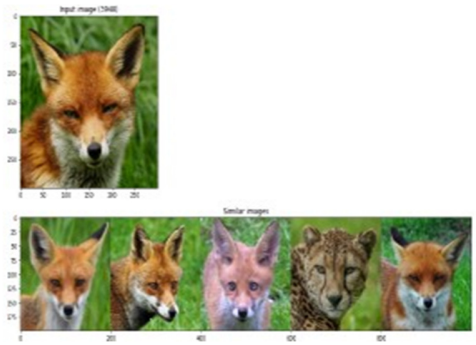

Fig. 6. Results with hyperparameter tuning (2nd image).

Fig. 7. Hyperparameter tuning (3rd image).

Fig. 8. Hyperparameter tuning (3rd image).

VIII. CONCLUSION

The study demonstrates that modern neural networks, despite their longer training times, exhibit remarkable capabilities. Using platforms such as Google Colab to access GPU and TPU resources significantly accelerates computational tasks, with image processing times averaging approximately 55ms (TP90) per picture. Although there is some variability in processing time, depending on image complexity, optimizing batch size, hyperparameters, and network architecture can substantially enhance model performance and reduce training durations. In particular, the effectiveness of these optimizations depends on the specific characteristics of the dataset under study.

This study proposed a DNN autoencoder model to facilitate image similarity analysis. This approach outperformed traditional CNNs in assessing picture similarity due to its ability to learn rich feature representations. Moreover, the adoption of cosine similarity instead of the conventional Euclidean distance method improved the accuracy and efficiency of similarity analysis. Using these innovative techniques, the proposed model achieved superior results in image similarity assessment, paving the way for enhanced image processing and analysis capabilities in various domains.

ACKNOWLEDGMENT

The authors would like to thank the Connected Intelligence Research Group (CIRG) and RMC at the University of Southampton Malaysia.

REFERENCES

[1] S. Merugu, M. C. S. Reddy, E. Goyal, and L. Piplani, "Text Message Classification Using Supervised Machine Learning Algorithms," in *ICCCE 2018*, 2018, pp. 141–150, https://doi.org/10.1007/978-981-13-0212-1_15.

[2] R. Yadav and D. Singh, "Malware Detection and Analysis Tools," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 11, no. 11s, pp. 735–744, Sep. 2023.

[3] S. Appalaraju and V. Chaoji, "Image similarity using Deep CNN and Curriculum Learning." arXiv, Jul. 13, 2018, https://doi.org/10.48550/arXiv.1709.08761.

[4] J. Wang *et al.*, "Learning Fine-Grained Image Similarity with Deep Ranking," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, USA, Jun. 2014, pp. 1386–1393, https://doi.org/10.1109/CVPR.2014.180.

[5] X. Gao *et al.*, "Sparse Online Learning of Image Similarity," *ACM Transactions on Intelligent Systems and Technology*, vol. 8, no. 5, May 2017, Art. no. 64, https://doi.org/10.1145/3065950.

[6] A. Dosovitskiy and T. Brox, "Generating Images with Perceptual Similarity Metrics based on Deep Networks," in *Advances in Neural Information Processing Systems*, 2016, vol. 29.

[7] F. Alotaibi, M. T. Abdullah, R. B. H. Abdullah, R. W. B. O. K. Rahmat, I. A. T. Hashem, and A. K. Sangaiah, "Optical Character Recognition for Quranic Image Similarity Matching," *IEEE Access*, vol. 6, pp. 554–562, 2018, https://doi.org/10.1109/ACCESS.2017.2771621.

[8] R. Zhang, L. Lin, R. Zhang, W. Zuo, and L. Zhang, "Bit-Scalable Deep Hashing With Regularized Similarity Learning for Image Retrieval and Person Re-Identification," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 4766–4779, Sep. 2015, https://doi.org/10.1109/TIP.2015.2467315.

[9] A. Wu, A. J. Piergiovanni, and M. S. Ryoo, "Model-based Behavioral Cloning with Future Image Similarity Learning," in *Proceedings of the Conference on Robot Learning*, May 2020, pp. 1062–1077.

[10] S. Jain and S. Shrivastava, "A novel approach for image classification in Content based image retrieval using support vector machine," *International Journal of Computer Science & Engineering Technology*, vol. 4, no. 3, pp. 223–227, Mar. 2013.

[11] S. Sajini and B. Pushpa, "A Binary Object Detection Pattern Model to Assist the Visually Impaired in Detecting Normal and Camouflaged Faces," *Engineering, Technology & Applied Science Research*, vol. 14, no. 1, pp. 12716–12721, Feb. 2024, https://doi.org/10.48084/etasr.6631.

[12] M. N. Saqib, H. Dawood, A. Alghamdi, and H. Dawood, "Weber's Law-based Regularization for Blind Image Deblurring," *Engineering, Technology & Applied Science Research*, vol. 14, no. 1, pp. 12937–12943, Feb. 2024, https://doi.org/10.48084/etasr.6576.

[13] W. Shimoda and K. Yanai, "Learning Food Image Similarity for Food Image Retrieval," in *2017 IEEE Third International Conference on Multimedia Big Data (BigMM)*, Laguna Hills, CA, USA, Apr. 2017, pp. 165–168, https://doi.org/10.1109/BigMM.2017.73.

[14] X. Yuan, Q. Liu, J. Long, L. Hu, and Y. Wang, "Deep Image Similarity Measurement Based on the Improved Triplet Network with Spatial Pyramid Pooling," *Information*, vol. 10, no. 4, Apr. 2019, Art. no. 129, https://doi.org/10.3390/info10040129.

[15] S. Merugu, K. Jain, A. Mittal, and B. Raman, "Sub-scene Target Detection and Recognition Using Deep Learning Convolution Neural Networks," in *ICDSMLA 2019*, 2019, pp. 1082–1101, https://doi.org/10.1007/978-981-15-1420-3_119.

[16] R. Sharma and A. Vishvakarma, "Retrieving Similar E-Commerce Images Using Deep Learning." arXiv, Jan. 11, 2019, https://doi.org/10.48550/arXiv.1901.03546.

[17] H. R. Tavakoli, A. Borji, J. Laaksonen, and E. Rahtu, "Exploiting inter-image similarity and ensemble of extreme learners for fixation prediction using deep features," *Neurocomputing*, vol. 244, pp. 10–18, Jun. 2017, https://doi.org/10.1016/j.neucom.2017.03.018.

[18] A. Stylianou, R. Souvenir, and R. Pless, "Visualizing Deep Similarity Networks," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Waikoloa, HI, USA, Jan. 2019, pp. 2029–2037, https://doi.org/10.1109/WACV.2019.00220.

[19] Y. Song, C. J. Rosenberg, A. Y. T. Ng, and B. Chen, "Evaluating image similarity," US8831358B1, Sep. 09, 2014.