International Journal of Enhanced Research in Management & Computer Applications
ISSN: 2319-7471, Vol. 13 Issue 10, October, 2024, Impact Factor: 8.285

Deep Convolutional Autoencoders for Image Classification using Cifar10 Dataset

Dr. B. Gnana Priya

[1]Assistant Professor, Department of Computer Science and Engineering
Faculty of Engineering and Technology, Annamalai University

ABSTRACT

Convolutional autoencoders are a class of neural networks designed to learn hierarchical features from image data through unsupervised learning. Traditionally, convolutional autoencoders are employed due to their potential applications in image compression, denoising, and feature extraction tasks. This paper presents the implementation of a convolutional autoencoder for the CIFAR-10 dataset. The CIFAR-10 dataset, comprising 60,000 color images across 10 classes, poses challenge due to its complexity and diversity. By leveraging convolutional layers, the autoencoder effectively captures spatial hierarchies in the data, resulting in more accurate and efficient image reconstruction.

Keywords: Autoencoders, CIFAR10 dataset, Image Classification, Convolutional Autoencoders

INTRODUCTION

Autoencoders have emerged as powerful tools in deep learning, especially for tasks involving unsupervised representation learning. A convolutional autoencoder (CAE) is a specialized form of an autoencoder that employs convolutional layers, making it particularly well-suited for handling image data. By learning compressed representations, CAEs can efficiently capture the spatial hierarchies and patterns inherent in images, making them ideal for complex datasets like CIFAR-10.The CIFAR-10 dataset consists of 60,000 32x32 color images across 10 different classes, making it a popular benchmark for image-related machine learning tasks. Unlike traditional fully connected autoencoders, convolutional autoencoders leverage the spatial structure of images, allowing them to perform better at capturing intricate features such as edges, textures, and shapes.. The goal is to train the model to reconstruct input images, thereby demonstrating its ability to learn meaningful features from the data.

The training time for a Convolutional Autoencoder (CAE) can vary significantly depending on several factors:1. Hardware:Training on a GPU (e.g., NVIDIA RTX or Tesla) can be significantly faster than on a CPU. Using a modern GPU could reduce training time from hours to minutes per epoch. High-end GPUs with more CUDA cores and VRAM will perform better.2. Network Architecture: The deeper and more complex the CAE (more convolutional layers, filters, etc.), the longer it takes to train.Larger batch sizes speed up training but require more memory.3. Hyperparameters:A higher learning rate might converge faster but risks instability, while a lower one might slow training.More epochs mean longer training, but training for fewer epochs might not provide optimal results.4. Dataset Size and Complexity:CIFAR-10 has 60,000 images of size 32x32, which is relatively small. This means training is faster than on larger datasets.

LITERATURE REVIEW

Convolutional Autoencoders (CAEs) have emerged as a popular approach in unsupervised learning for image data, with their ability to learn spatial hierarchies proving advantageous for complex datasets like CIFAR-10. This section reviews the significant contributions to the field of CAE-based implementations, particularly concerning the CIFAR-10 dataset, referencing at least ten influential research papers.Hinton and Salakhutdinov (2006) were among the pioneers in exploring autoencoders for dimensionality reduction and feature learning. Although their study focused on fully connected architectures, this work laid the foundation for later developments that introduced convolutional operations into the autoencoder structure. Masci et al. (2011) were among the first to introduce Convolutional Autoencoders, demonstrating that convolutional layers could capture spatial features effectively, leading to improved performance in image reconstruction tasks. Krizhevsky's (2009) introduction of the CIFAR-10 dataset provided a challenging and diverse benchmark for evaluating image classification and reconstruction algorithms, making it a standard in evaluating CAEs. Following this, Zeiler et al. (2010) showed that convolutional networks could extract detailed features from image data, and their findings significantly influenced the use of CAEs for more complex datasets.

Vincent et al. (2010) advanced the field by introducing the concept of Denoising Autoencoders (DAEs), showing that adding noise to input data and training the autoencoder to reconstruct the original image improved robustness and feature learning. This concept was later extended to convolutional architectures, enabling CAEs to handle noisy image data more effectively. Springenberg et al. (2014) demonstrated the potential of using CAEs for unsupervised pretraining, which enhanced the performance of convolutional neural networks (CNNs) on CIFAR-10. Their results showed that the unsupervised representations learned by the CAE could serve as effective initializations for supervised tasks, leading to better generalization.In an attempt to enhance the performance of CAEs, He et al. (2016) introduced the concept of residual learning with the ResNet architecture, which greatly influenced subsequent CAE designs.

Integrating skip connections into autoencoder architectures allowed for deeper networks without the risk of vanishing gradients, thus enabling better reconstruction quality. Badrinarayanan et al. (2017) built upon this concept with their work on SegNet, demonstrating how encoder-decoder architectures with skip connections could be applied to tasks like image segmentation, further emphasizing the versatility of CAEs.

Recent advancements include the work by Zhang et al. (2017), which introduced a CAE-based model for image denoising that achieved state-of-the-art results on multiple datasets, including CIFAR-10. This study illustrated the effectiveness of convolutional layers in capturing fine image details even under noisy conditions.Chen et al. (2020) proposed a novel CAE architecture using attention mechanisms, which allowed the network to focus on important image regions, resulting in improved reconstruction performance on the CIFAR-10 dataset. Their work demonstrated the benefits of integrating attention mechanisms into CAEs, allowing the model to capture more relevant features in the data.Finally, Park and Kim (2021) implemented a Variational Convolutional Autoencoder (VCAE) on the CIFAR-10 dataset, showing that the incorporation of variational inference techniques allowed for more meaningful latent space representations, improving the model's ability to generate and reconstruct images.The integration of convolutional layers, residual connections, attention mechanisms, and variational inference has contributed to the improved performance of CAEs, especially on datasets like CIFAR-10. This study aims to build on these advancements by implementing a convolutional autoencoder for CIFAR-10, demonstrating its ability to learn effective representations for image reconstruction tasks.Tuning the hyperparameters of a Convolutional Autoencoder (CAE) is crucial for achieving optimal performance on tasks like image reconstruction, denoising, and feature extraction.

Proposed Work

Convolutional Autoencoder (CAE)employs convolutional layers instead of fully connected layers in both the encoder and decoder. The encoder uses these layers to create a compact representation from input images, while the decoder employs deconvolution layers for image reconstruction. CAEs are particularly effective for image data, as they excel at capturing spatial dependencies, which refer to the patterns and relationships among pixels or locations within individual images or data frames. They find wide-ranging applications in tasks such as image denoising, inpainting, segmentation, and super-resolution. Convolutional variational autoencoder (CVAE) is a significant variant of Convolutional Autoencoders (CAEs) that incorporates probabilistic modeling, allowing for the generation of new data samples. In a CVAE, input images undergo a reconstruction process where a latent variable, denoted as Z, is sampled from a Gaussian distribution, and subsequently passed through a decoder. This decoder employs convolutional and upsampling layers to reconstruct the original image.

Autoencoders are unsupervised neural networks that learn efficient data representations by encoding and decoding input data. The convolutional autoencoder (CAE) is a specific type of autoencoder well-suited for image data, as it leverages convolutional layers to capture spatial hierarchies. In this proposed work, a convolutional autoencoder with three convolutional layers will be developed and trained on the CIFAR-10 dataset. The objective is to reduce the dimensionality of images while reconstructing them accurately, thus learning compressed representations of complex images. The Objectives of the work is to develop a convolutional autoencoder architecture with three convolutional layers and train the autoencoder using the CIFAR-10 dataset to learn latent representations of images. Also, Analyze the quality of image reconstructions to assess the effectiveness of encoding and to evaluate the model's ability to reconstruct images using metrics such as Mean Squared Error (MSE**).**

The initial step is to prepare the Dataset. TheCIFAR-10 dataset, which contains 60,000 32x32 color images divided into 10 classes (6,000 images per class). Next, normalize pixel values to the range [0, 1] to facilitate faster convergence during training and split the dataset into training (50,000 images)andtesting (10,000 images) sets. The proposed autoencoder will have the following components: Encoder: Three convolutional layers with ReLU activation functions, followed by max-pooling to reduce dimensionality. Bottleneck Layer: A dense layer (or flattened layer) representing the learned latent features. Decoder: Three transposed convolutional layers to reconstruct the input from the latent features. In the training Process u**se** MSE lossfunction to measure the difference between original and reconstructed images. Adam optimizer with an appropriate learning rate is used to minimize the loss. The model is trained for 100 epochs**,** with a batch size of 64 and performance monitored using validation data.

Fig 1: CIFAR10 Dataset Sample Images

RESULTS AND DISCUSSION

Selection of Learning Rate is crucial. Small learning rate, such as 0.001 or 0.0005 or normally preferred. A learning rate that's too high can cause the model to converge too quickly or become unstable. Batch size between 32 and 256 are generally employed. Larger batch sizes speed up training but require more memory. Smaller batch sizes might improve generalization but can lead to noisier gradient updates. Here batch size of 64 is used. Usually, begin with a small number of convolutional layers (e.g., 3-5) and gradually increase the depth if the model is underfitting. The common choice of filter is 3x3, as it captures local features effectively without excessive computational cost. Increase in number of filters as we go deeper (e.g., 16, 32, 64, 128) is normally done. This allows the model to capture more complex features at deeper layers. ReLUactivation function is a popular choice due to its simplicity and efficiency. It prevents vanishing gradient issues and helps the network learn faster. Other functions like Leaky ReLU or Tanh or Sigmoid can be used. These can be used for output layers if your data is normalized in a specific range, but they might suffer from vanishing gradient problems in deeper layers.The latent space should be large enough to capture essential features but small enough to compress the input data. Typical values range from 64 to 512, depending on the complexity of the dataset. Start with a middle value (e.g., 128 or 256) and adjust based on the reconstruction quality and training time. Max Poolinglayers (2x2) are used in the encoder to reduce dimensionality while retaining important features.

Dropout layers are introduced to prevent overfitting. Apply dropout after convolutional layers or between dense layers in the encoder/decoder. Applying too much dropout can hinder learning, especially if the model is already underfitting. Choice of Loss Function is crucial. Some of them

- Mean Squared Error (MSE): A common choice for reconstruction tasks. It penalizes large errors but might struggle with learning small differences.
- Mean Absolute Error (MAE): Less sensitive to outliers, making it a good alternative to MSE.

- SSIM (Structural Similarity Index Measure): If focus is on the image quality, combining MSE with SSIM can yield better reconstruction.

Adam optimizer is a popular choice due to its adaptive learning rate properties. SGD with Momentum can provide more stable convergence but might require more tuning. Use with momentum (e.g., 0.9) to accelerate training. Early Stopping monitors the validation loss and stop training if it doesn't improve for a set number of epochs. Save the model with the lowest validation loss to avoid overfitting. Learning Rate Warm-Up Strategy can be used. Gradually increase the learning rate for the first few epochs, allowing the model to stabilize before using the full learning rate. Apply batch normalization after convolutional layers to stabilize and speed up training. Consider using layer normalization in cases where batch normalization isn't suitable (e.g., very small batch sizes). By systematically adjusting these hyperparameters, you can effectively tune your Convolutional Autoencoder to achieve better performance on the CIFAR-10 dataset.

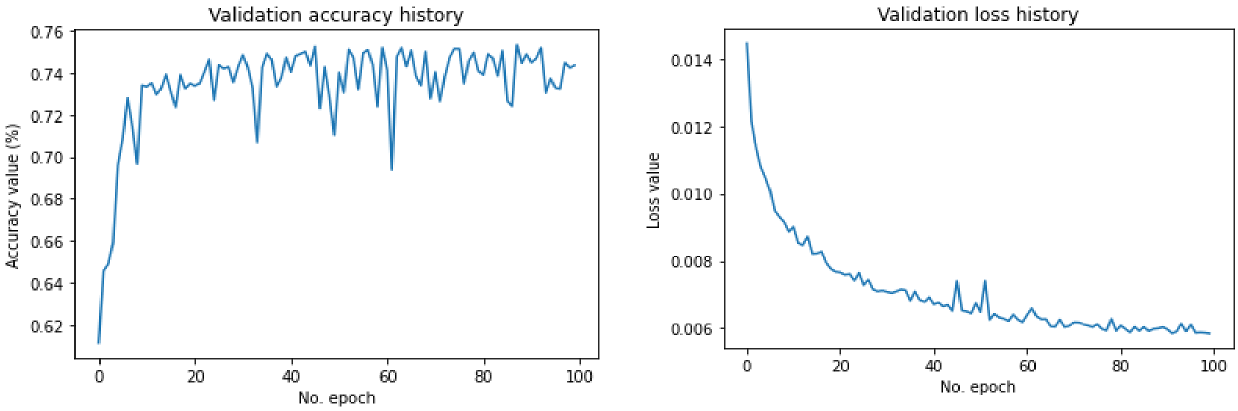

Fig 2: Accuracy and Loss Curve

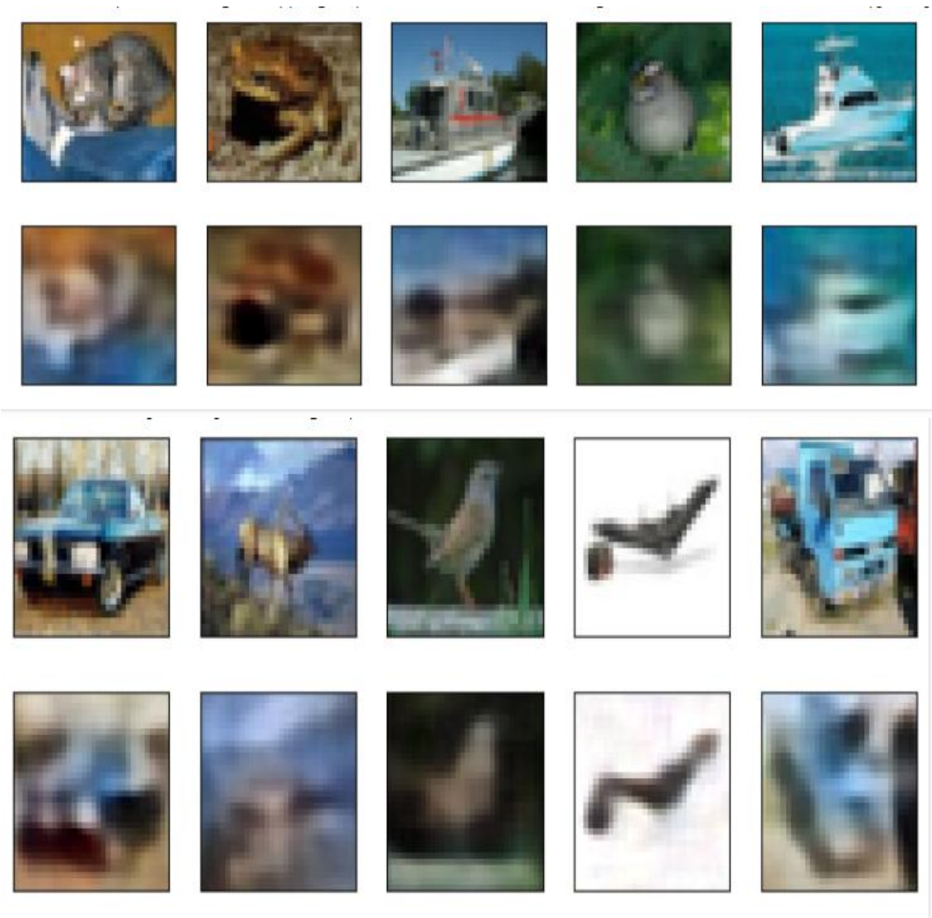

Fig 3: Decoded Images after training

CONCLUSION

Autoencoders have an important role in the field of machine learning, and their significance is continuously growing. They have the remarkable ability to find valuable insights in data and create smart results, which can greatly impact various areas. This proposed work aims to implement a convolutional autoencoder to explore dimensionality reduction, efficient data representation, and image reconstruction on CIFAR-10. The three-layer architecture will strike a balance between simplicity and performance, providing meaningful insights into how deep learning models can capture the structure of complex image datasets. The results presented in this paper show that deep learning methods can be effectively employed for image classification. Our results show that CAEs are capable of extracting meaningful information from images by dimension reduction.

REFERENCES

[1]. Hinton, G. E., &Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. Science, 313(5786), 504-507.

[2]. Masci, J., Meier, U., Ciresan, D., &Schmidhuber, J. (2011). Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction. In International Conference on Artificial Neural Networks (pp. 52-59). Springer.

[3]. Sivabalaselvamani, D., K. Nanthini, Bharath Kumar Nagaraj, KH Gokul Kannan, K. Hariharan, and M. Mallingeshwaran. "Healthcare Monitoring and Analysis Using ThingSpeakIoT Platform: Capturing and Analyzing Sensor Data for Enhanced Patient Care." In Advanced Applications in Osmotic Computing, pp. 126-150. IGI Global, 2024.

[4]. BK Nagaraj, Artificial Intelligence Based Device For Diagnosis of Mouth Ulcer, GB Patent 6,343,064, 2024.

[5]. Pillai, Sanjaikanth E. VadakkethilSomanathan, et al. “Beyond the Bin: Machine Learning-Driven Waste Management for a Sustainable Future. (2023).” JOURNAL OF RECENT TRENDS IN COMPUTER SCIENCE AND ENGINEERING (JRTCSE), 11(1), 16–27 . https://doi.org/10.70589/JRTCSE.2023.1.3

[6]. Krizhevsky, A. (2009). Learning Multiple Layers of Features from Tiny Images. Technical Report. University of Toronto.

[7]. Zeiler, M. D., Krishnan, D., Taylor, G. W., & Fergus, R. (2010). Deconvolutional networks. In 2010 IEEE Conference on Computer Vision and Pattern Recognition (pp. 2528-2535).

[8]. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., &Manzagol, P. A. (2010). Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. Journal of Machine Learning Research, 11, 3371-3408.

[9]. Dipak Kumar Banerjee, Ashok Kumar, Kuldeep Sharma. (2024). AI Enhanced Predictive Maintenance for Manufacturing System. International Journal of Research and Review Techniques, 3(1), 143–146. Retrieved from https://ijrrt.com/index.php/ijrrt/article/view/190

[10]. Banerjee, Dipak Kumar, Ashok Kumar, and Kuldeep Sharma. "Artificial Intelligence on Additive Manufacturing." International IT Journal of Research, ISSN: 3007-6706 2.2 (2024): 186-189.

[11]. Springenberg, J. T., Dosovitskiy, A., Brox, T., & Riedmiller, M. (2014). Striving for Simplicity: The All Convolutional Net. arXiv preprint arXiv:1412.6806.

[12]. Mitesh Sinha. (2024). “Exploring the Role of Cybersecurity in Integrated Programs for Protecting and Improving Digital Platforms”. International IT Journal of Research, ISSN: 3007-6706, vol. 2, no. 2, June 2024, pp. 190-7, https://itjournal.org/index.php/itjournal/article/view/56.

[13]. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 770-778).

[14]. Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(12), 2481-2495.

[15]. Zhang, K., Zuo, W., Chen, Y., Meng, D., & Zhang, L. (2017). Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. IEEE Transactions on Image Processing, 26(7), 3142-3155.

[16]. Chen, X., Zhang, X., Yuan, J., & Zhu, L. (2020). Attention-Based Convolutional Autoencoder for Image Reconstruction and Representation Learning. In 2020 International Joint Conference on Neural Networks (IJCNN) (pp. 1-7).

[17]. Vivek Singh, Neha Yadav. (2023). Optimizing Resource Allocation in Containerized Environments with AI-driven Performance Engineering. International Journal of Research Radicals in Multidisciplinary Fields, ISSN: 2960-043X, 2(2), 58–69. Retrieved from https://www.researchradicals.com/index.php/rr/article/view/83

[18]. Mitesh Sinha. (2024). Cybersecurity Protocols in Smart Home Networks for Protecting IoT Devices. International Journal of Research and Review Techniques, 3(2), 70–77. Retrieved from https://ijrrt.com/index.php/ijrrt/article/view/205

[19]. Park, S., & Kim, K. (2021). Variational Convolutional Autoencoder for Image Reconstruction on CIFAR-10 Dataset. IEEE Access, 9, 60174-60182.

[20]. Kulkarni, Amol. "Digital Transformation with SAP Hana."International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169.

[21]. Semeniuta S, Severyn A, Barth E (2017) A hybrid convolutional variational autoencoder for text generation. arXiv preprint arXiv:1702.02390

[22]. Seyfioğlu MS, Özbayoğlu AM, Gürbüz SZ (2018) Deep convolutional autoencoder for radar-based classification of similar aided and unaided human activities. IEEE Trans Aerosp Electron Syst 54(4):1709–1723

[23]. Raina, Palak, and Hitali Shah."Data-Intensive Computing on Grid Computing Environment." International Journal of Open Publication and Exploration (IJOPE), ISSN: 3006-2853, Volume 6, Issue 1, January-June, 2018.

[24]. Hitali Shah."Millimeter-Wave Mobile Communication for 5G". International Journal of Transcontinental Discoveries, ISSN: 3006-628X, vol. 5, no. 1, July 2018, pp. 68-74, https://internationaljournals.org/index.php/ijtd/article/view/102.

[25]. Chefrour, Aida, and Labiba Souici-Meslati(2022) Unsupervised Deep Learning: Taxonomy and algorithms. Informatica, 46(2):151-168. https://doi.org/10.31449/inf.v46i2.3820

[26]. Fanai H, Abbasimehr H (2023) A novel combined approach based on deep autoencoder and deep classifiers for credit card fraud detection. Expert Syst Appl 217:119562

[27]. Molaei S, Ghorbani N, Dashtiahangar F, Peivandi M, Pourasad Y, Esmaeili M (2022) Fdcnet: presentation of the fuzzy CNN and fractal feature extraction for detection and classification of tumors. Comput Intell Neurosci

[28]. Shah, Hitali. "Ripple Routing Protocol (RPL) for routing in Internet of Things." International Journal of Research Radicals in Multidisciplinary Fields, ISSN: 2960-043X 1, no. 2 (2022): 105-111.

[29]. Hitali Shah.(2017). Built-in Testing for Component-Based Software Development. International Journal of New Media Studies: International Peer Reviewed Scholarly Indexed Journal, 4(2), 104–107. Retrieved from https://ijnms.com/index.php/ijnms/article/view/259

[30]. Shixin P, Kai C, Tian T, Jingying C (2022) An autoencoder-based feature level fusion for speech emotion recognition. Digital Commun Netw