*Exploring Hadoop Distributed File System (HDFS) and Advanced Neural Networks for Scalable Image Data Analysis: A CIFAR-10 Study*

C McKenna   
College of Computing Technology  
Dublin, Ireland  
Email: sbs23014@student.cct.ie

This research delves into the utilization of Hadoop Distributed File System (HDFS) as a big data storage solution and its integration with advanced neural networks, specifically Convolutional Neural Networks (CNNs) equipped with sequencers. The installation and setup of Hadoop were found to pose initial challenges and revealed its user-friendliness lagging behind more recent tools. Despite these challenges, the neural network performed reasonably well, achieving an accuracy rate of 53%. In the context of this widely explored topic, where numerous options and research papers exist, this study aims to contribute valuable insights into the seamless integration of scalable data storage solutions with cutting-edge deep learning techniques

# Introduction

# In the realm of modern data science and machine learning, the analysis of massive datasets have become quintessential for obtaining insights and developing advanced predictive and classification models. The advent of big data technologies has ushered in a new era where the storage and processing of large data volumes is not only feasible but essential for solving complex problems across various domains. Within this context, the choice of an appropriate data storage system plays a pivotal role in enabling efficient data management, retrieval, and analysis. Alongside this, the rapid advancements in deep learning have led to the emergence of complex neural network architectures, including Convolutional Neural Networks (CNNs), which have demonstrated impressive ability in media classification, including classification of images.

# This research paper looks to explore some big data storage solutions, with a particular focus on the Hadoop Distributed File System (HDFS). At the same time, it analyses the use of advanced neural network models, particularly CNNs equipped with sequencers, for the classification of a widely studied dataset - CIFAR-10. Through this analysis, it aims to provide valuable insights into the scalable data storage solutions as well as cutting-edge deep learning techniques for the efficient processing and analysis of large-scale image datasets.

## Big Data Storage Options

In today's data-driven world, managing and storing vast datasets pose significant challenges. Traditional storage solutions often fall short in terms of scalability, fault-tolerance, and cost-effectiveness when dealing with big data. HDFS is a distributed file system designed explicitly to tackle these challenges. HDFS partitions large files into smaller blocks, distributes them across a cluster of machines, and replicates them to ensure fault tolerance. Its robust architecture makes it a popular choice for storing and managing large datasets, making it a core component of the Hadoop ecosystem.

In this paper, we delve into the core features of HDFS and examine its role in the efficient storage of large-scale datasets.

## Deep Learning with CNN Sequencers

In recent years, deep learning has revolutionised the field of machine learning, especially in image-related tasks. While some argue that CNNs cannot handle very large images, they are still a very popular choice (Das et al., 2020), showcasing impressive capabilities in image classification, object detection, and image generation. Moreover, CNNs can be further enhanced with sequencers, allowing them to capture temporal dependencies in sequential data, opening doors to sophisticated tasks such as video analysis, speech recognition, and sequential image classification.

In this research, the design and implementation of CNN sequencers is investigated, along with their application to image classification tasks.

# Dataset

The CIFAR-10 dataset, short for the Canadian Institute For Advanced Research - 10, was created by Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton in 2009. It consists of 60,000 32x32 color images spread across 10 distinct classes, with each class containing 6,000 images. These classes are:

Airplane

Automobile

Bird

Cat

Deer

Dog

Frog

Horse

Ship

Truck

The dataset is divided into two subsets: a training set with 50,000 images and a testing set with 10,000 images, making it suitable for both training and evaluation purposes. One of the key characteristics of CIFAR-10 is that it contains real-world, low-resolution images, which poses several unique challenges and benefits for CNN-based research.

## Advantages for CNN Perspective:

**Small Image Size:** The 32x32 pixel resolution of CIFAR-10 images is relatively small compared to many other datasets in computer vision. This small size makes it computationally efficient, allowing researchers to experiment with CNN architectures more quickly.

**Diversity of Classes:** CIFAR-10 encompasses a diverse range of object classes, including animals, vehicles, and more. This diversity encourages the development of models that can handle a wide array of visual concepts, making it valuable for general-purpose image classification tasks.

**Real-World Complexity**: The images in CIFAR-10 are more complex and varied compared to synthetic or highly curated datasets. This realism is crucial for training CNNs to work effectively in real-world scenarios, where images are often noisy, contain various backgrounds, and exhibit different lighting conditions.

**Comparative Benchmark**: CIFAR-10 has become a standard benchmark in the field of computer vision. Many state-of-the-art CNN architectures and techniques are evaluated on this dataset. This makes it easier to compare and benchmark the performance of new CNN models against existing ones, facilitating progress in the field.

**Accessibility:** CIFAR-10 is freely available for research purposes, making it accessible to a wide range of researchers and students. Its availability has contributed to its popularity in the machine learning community.

The CIFAR-10 dataset is a valuable resource for researchers working with CNNs. Its small image size, diverse classes, real-world complexity, limited data, and status as a benchmark dataset make it an excellent choice for exploring and advancing CNN-based computer vision research. Researchers often use CIFAR-10 as a stepping stone to develop and fine-tune their models before applying them to more challenging and extensive datasets, making it an integral part of the computer vision research landscape.

# Literature Review

Different researchers have provided many solutions to the storage and processing of image datasets such as the Cifar10 dataset. One survey was carried out in 2016 (Lim, Young and Patton, 2016), which looked to employ the caffe deep learning framework to train neural network models against the well-known MINST, CIFAR10 and ImageNet. It could be assumed that one reason the researchers choose the Caffe framework is because it is a well-established deep learning framework and is relatively easy to use (Majumder et al., 2017). Lim, Young and Patton analysed the effectiveness of the following systems:

(1) PNG-formatted image files on local file system

(2) Pushing pixel arrays from image files into a single HDF5 file on local file system;

(3) In-memory arrays to hold the pixel arrays in Python and C++;

(4) Loading the training data into LevelDB, a log-structured merge tree based key-value storage; and

(5) Loading the training data into LMDB, a B+tree based keyvalue storage

The paper successfully highlights the inefficiencies of using image files that are saved locally on systems such as ext4 – the study notes that training can be up to 17 times slower. It could be regarded as unfortunate that the researchers didn’t extend the survey to encompass more efficient systems such as Amazon S3, Google Cloud Storage or HDFS which is being used for this research paper.

The paper also concludes that to effectively train deep neural networks, it is common to use large datasets that are accessed through mini-batches during training. However, traditional file systems are not well-suited for this task due to their inefficient indexing and caching mechanisms. Instead, they recommend utilizing image storage backends with efficient indexing capabilities for training samples and improved caching mechanisms. This approach will enhance the training process of deep neural networks by optimizing data access and retrieval.

**PHDFS: Optimizing I/O performance of HDFS in deep learning cloud computing platform (Zhu et al., 2020)**

This research paper discusses the significance of the file system in deep learning cloud computing platforms, highlighting the use of Hadoop Distributed File System (HDFS) in large-scale clusters due to its high performance and availability. However, it points out a critical issue in deep learning datasets where the number of files is extensive, but individual file sizes are small, leading to performance problems in HDFS.

To address this issue, the paper introduces a new approach called Pile-HDFS (PHDFS), which focuses on aggregating small files based on their correlation. The concept of "Pile" is introduced as an I/O unit that combines groups of small files with similar characteristics. Additionally, the research proposes a two-layer manager and incorporates internal organization information into data blocks to improve small file access efficiency.

The experimental results presented in the paper show that PHDFS outperforms the original HDFS significantly. It reduces latency when accessing small files and enhances the Frames Per Second (FPS) of typical deep learning models by up to 40%. This suggests that PHDFS offers a valuable solution to the performance challenges associated with deep learning datasets in cloud computing environments.

# Methodology

We are performing a data pipeline for deep learning with the CIFAR-10 dataset. First, the CIFAR-10 dataset is saved into the Hadoop Distributed File System (HDFS), which is a distributed and scalable storage system. This step allows us to efficiently store and manage the dataset in a distributed computing environment.

Next, CIFAR-10 dataset was imported into Python using the Pickle module. Pickle is a Python library that allows us to serialize and deserialize Python objects, making it a convenient choice for loading and working with datasets.

The meta data is brought in, again from HDFS and the data is visualized using the Matplotlib library to ensure it was loaded correctly. The low resolution of the images is very evident here.

Finally, we are running a neural network using the imported CIFAR-10 dataset. This neural network is designed for tasks such as image classification. The neural network will be trained on this data to learn patterns and features within the images, enabling it to make predictions or classifications based on new, unseen images.

In summary, our workflow involves storing the CIFAR-10 dataset in HDFS for efficient storage, importing it into Python using Pickle for data manipulation, and then training a neural network on this dataset for image-related tasks.

# Results and discussion

After training the CNN model on the CIFAR-10 dataset for 150 epochs, the following results were obtained:

1. Accuracy: 53%

2. Loss: Approximately 1.33

The achieved accuracy of 53% indicates that the model was able to correctly classify approximately 53% of the images in the test dataset. While this accuracy is better than random guessing (10% for a 10-class problem), it may not be considered high for practical applications. Indeed other research papers following other methodologies achieved better results.

Factors contributing to this accuracy include the complexity of the CIFAR-10 dataset, the architecture of the neural network, and the training parameters. CIFAR-10 consists of 60,000 32x32 color images across 10 classes, making it a challenging benchmark for image classification tasks.

Potential areas for improvement in accuracy include experimenting with different architectures, hyperparameters, and regularization techniques. Increasing the depth of the CNN, adjusting learning rates, and implementing techniques like dropout or batch normalization can potentially enhance accuracy.

Efforts were made to enhance the accuracy of the neural network by implementing GridSearchCV, a powerful hyperparameter optimization technique. However, due to the complexity of the task and limited available time, configuring GridSearchCV effectively proved to be a challenge. Despite the attempt, its full potential in optimizing the neural network's performance remained unexplored within the allotted timeframe.

The loss value of approximately 1.33 indicates the error or mismatch between the predicted and actual labels during training. Lower loss values are generally desirable, as they signify a better fit of the model to the training data. To reduce the loss further, one could explore different optimization algorithms, such as stochastic gradient descent with momentum (SGD) or adaptive learning rate methods like Adam. Again, fine-tuning the model's hyperparameters, using methods such as GridsearchCV, and using data augmentation can help in minimizing the loss.

Data augmentation was applied to the training images using techniques such as shifting, rotation, and flipping. This helped to increase the diversity of the training data, which can lead to better generalisation. Data augmentation is a valuable tool for improving model performance, especially when the dataset is limited in size.

The model was for 10, 50 and 100 epochs and this was further increased to 150 because a slight accuracy improvement was found with each epoch. However after 150 epochs, the accuracy did not seem to improve.

# Conclusion

In conclusion, this research journey into training a neural network on the CIFAR-10 dataset has yielded promising results, but it also highlights the vast opportunities for further advancements. The achieved accuracy of approximately 53% demonstrates the potential of our model, especially given the dataset's intricacies. However, a lot more work could be carried out to improve the accuracy and reduce the loss function.

1. Lim, S.-H., Young, S.R. and Patton, R.M. (2016). An analysis of image storage systems for scalable training of deep neural networks.
2. Majumder, U., Christiansen, E., Wu, Q., Inkawhich, N., Blasch, E. and Nehrbass, J. (2017). High-performance computing for automatic target recognition in synthetic aperture radar imagery. Cyber Sensing 2017. doi:https://doi.org/10.1117/12.2263218.
3. Das, K., Conjeti, S., Chatterjee, J. and Sheet, D. (2020). Detection of Breast Cancer From Whole Slide Histopathological Images Using Deep Multiple Instance CNN. IEEE Access, 8, pp.213502–213511.
4. Zhu, Z., Tan, L., Li, Y. and Ji, C. (2020). PHDFS: Optimizing I/O performance of HDFS in deep learning cloud computing platform. Journal of Systems Architecture, [online] 109, p.101810.
5. Lu, S., Wei, X., Li, Y. and Wang, L. (2018). Detecting Anomaly in Big Data System Logs Using Convolutional Neural Network. Dependable Autonomic and Secure Computing.
6. Zhang, H., Huang, H. and Wang, L. (2017). MRapid: An Efficient Short Job Optimizer on Hadoop. International Parallel and Distributed Processing Symposium.

*Dataset reference:*

Krizhevsky, A. (2009). Learning Multiple Layers of Features from Tiny Images

**IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove template text from your paper may result in your paper not being published.**