Boosting Machine Learning:

The Impact of Data Augmentation on Model Performance

Jaylon Nelson-Sellers

University of Washington Bothell

CSS 497

Professor Wooyoung Kim

6/7/24

Abstract *(for professional papers)*

This study investigates the effect of dataset dimensionality on the performance of machine learning (ML) and deep learning (DL) models, specifically when complex DL approaches outperform classic ML models such as Random Forests. It answers three questions: (1) when to employ DL instead of ML models, (2) the advantages of augmenting tabular data for DL, and (3) performance comparisons to determine the best generalist model.

Scalable Artificial Neural Networks (ANNs) were built and tested using a variety of datasets for classification and regression, including tabular, picture, and sequential data. ML models, including Decision Trees, Random Forests, K-nearest neighbors, ANNs, CNNs, LSTMs, and Transformers, were tested on eight datasets. Performance was assessed using mean square error, accuracy, and F1 score.

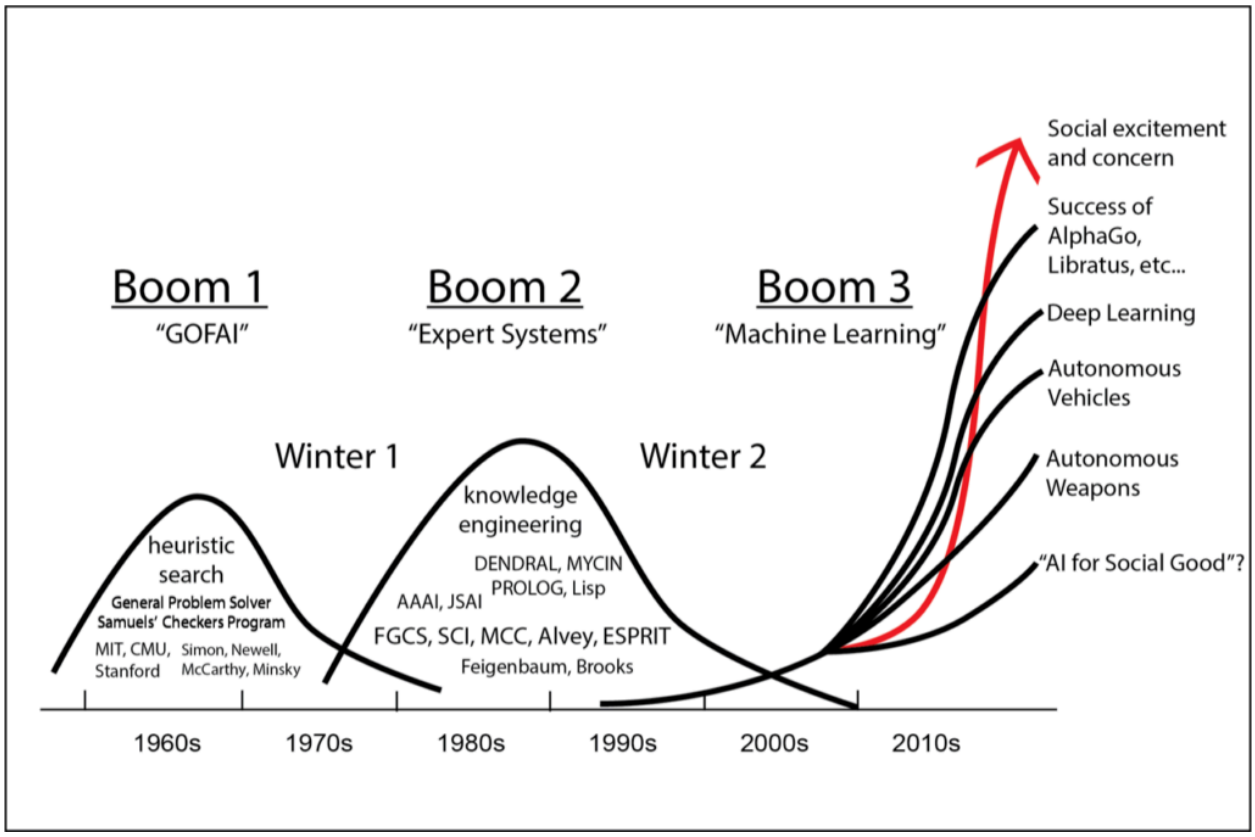
The results demonstrated that DL models favor naturally augmentable datasets in two dimensions (for example, photos or sequential data). For tabular data, augmenting to two dimensions and training using CNNs provided minimal advantage. Contrary to expectations, CNNs outperformed other models, whereas Transformers underperformed. Key findings: (1) DL models excel at image classification and sequential data forecasting; (2) Random Forests outperform DL models for most tabular data with less complexity; and (3) CNNs are best for image tasks, while Recurrent Networks are best for sequential data, though CNNs also perform well.

The study recommends ML and DL models based on dataset features, determining when DL models are useful and when regular ML models are sufficient.

Boosting Machine Learning: The Impact of Data Augmentation on Model Performance

Introduction

Machine Learning (ML) is a rapidly evolving field with substantial contributions from industry heavyweights such as NVIDIA and Tesla. Deep Learning (DL) is a specialized subset of machine learning that focuses on neural networks with numerous layers to solve challenging problems. Given the rising use of AI across industries, it is critical to understand which technologies are most suited for various jobs. This study aims to clarify the conditions in which classic ML models outperform DL models and vice versa.



Graph of AI Booms, with Boom 3 Representing the current trends.

Previously, while studying under Professor Kim, software was developed to help biologists adopt deep learning approaches to their specific research issues. This project entailed developing a software wrapper for PyTorch, a popular DL library in Python. PyTorch supports

CUDA, a parallel computing platform and programming style developed by NVIDIA that uses GPUs to improve computational speed. Although this experiment gave vital insights into the practical uses of deep learning, it also indicated that the medical datasets employed did not benefit greatly from DL approaches regarding performance improvement.

The current capstone project intends to explore deep learning and machine learning by studying multiple datasets and comparing model performance. The main question posed is when does DL surpass classical ML techniques?

Research Objectives

- Determine when DL models should be preferred over traditional ML models.
- Assess the advantages of augmenting data dimensions for DL models.
- Evaluate the performance of ML and DL models across diverse use cases.

Methodology

Models

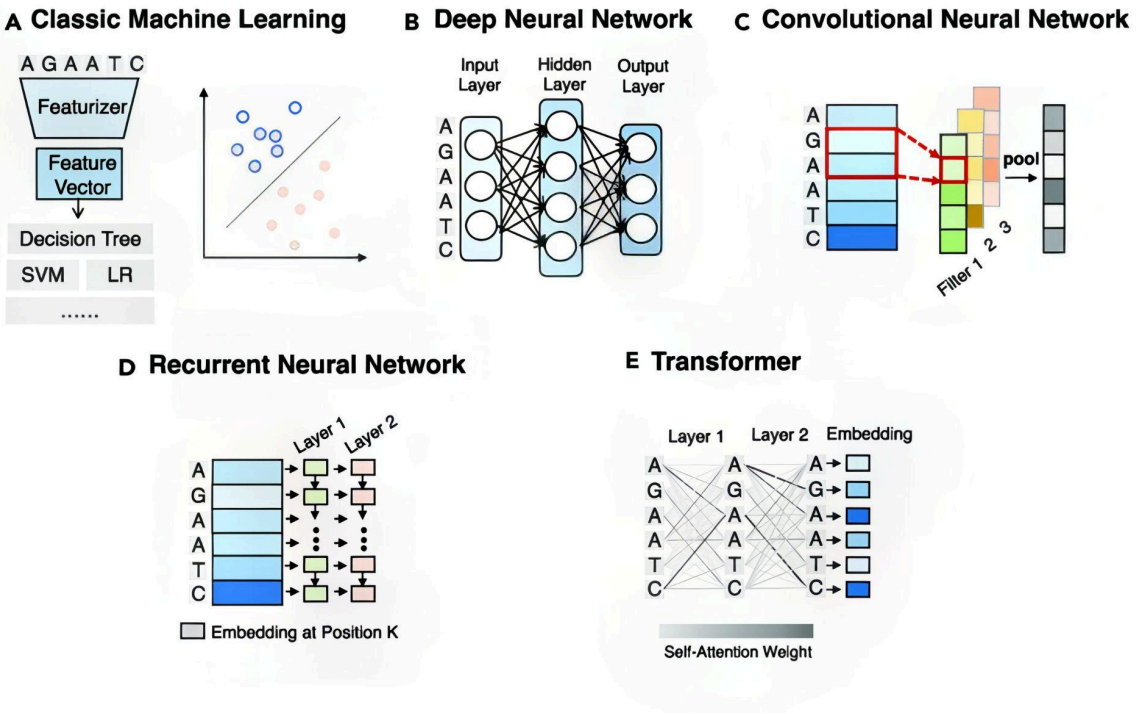
EasyTorch Development

The EasyTorch Library, which was created to help with the rapid building and deployment of Deep Learning (DL) models, serves as the project's core. EasyTorch acts as a wrapper around PyTorch, allowing easy interaction with Sci-Kit Learn modules. This integration facilitates the training, testing, and assessing of models by enabling the same training and predicting functionalities as the well-known Sci-Kit framework.

EasyTorch's development entailed automating several complicated components of DL model configuration. For example, setting up convolutional neural networks was simplified by limiting factors such as stride lengths and pooling to only 3-4 configurable options, making it

easier for users to experiment with these settings. While not the primary emphasis of this project, the completion of the EasyTorch Library considerably improved the efficiency and convenience of model construction, marking a critical milestone in the project's progress.

Machine Learning Models Used



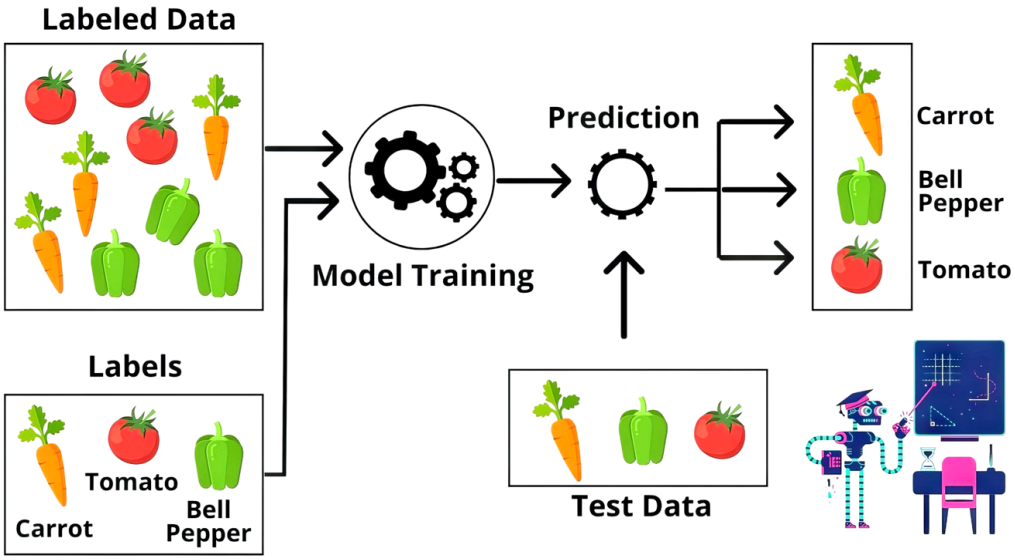
Example ML and DL models used this research

Various models were used in the study, each having its own set of strengths and applications. Decision Trees, a non-parametric supervised learning method, were used for classification and regression problems, with data divided into branches depending on feature thresholds. K-Nearest Neighbors (KNN) is an instance-based learning technique that classifies data points based on the majority class of their k-nearest neighbors in the feature space. Random Forests, an ensemble learning method, combines numerous decision trees to improve forecast accuracy while minimizing overfitting.

Deep Learning Models Used

Transformers, a Deep Learning (DL) model architecture, used self-attention techniques to efficiently analyze sequential input, excelling especially at natural language processing tasks. Long Short-Term Memory (LSTM) networks, a recurrent neural network, were created to model sequences and time-series data, efficiently capturing long-term dependencies while avoiding the vanishing gradient problem. General Neural Networks, algorithms fashioned after the human brain, were utilized to perform tasks such as classification, regression, and clustering using interconnected nodes. Finally, Convolutional Neural Networks (CNNs), designed for processing grid-like topology data such as photographs, use convolutional layers to capture spatial hierarchies and patterns.

Datasets

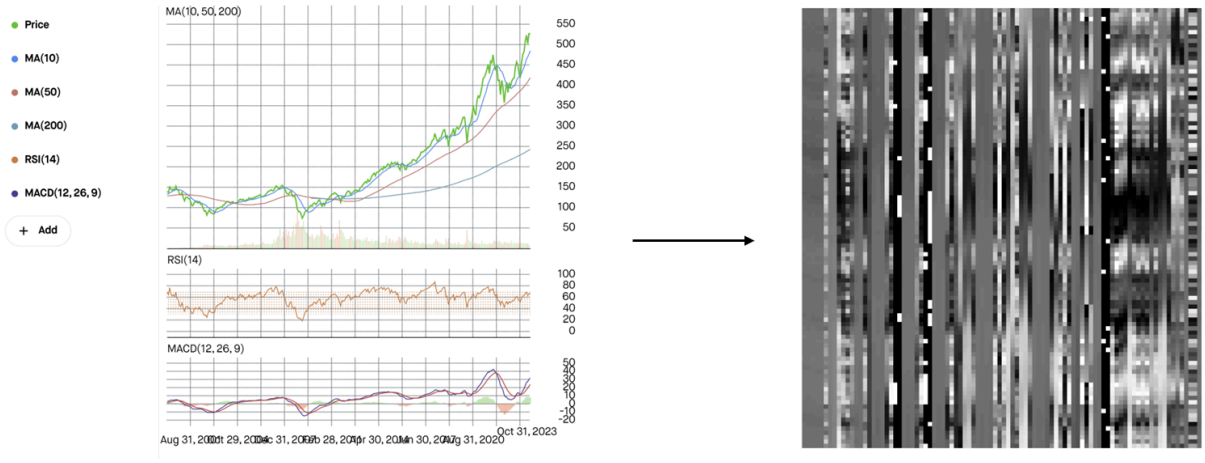


Graphic Showing the Classification Training Process

All datasets utilized in this project are classification problems, intending to correctly categorize each data sample into the right group. The study used eight datasets, split evenly

between stock and picture data. These datasets were chosen to provide a wide range of difficulties for the models while also allowing for significant comparisons with other studies in the field.

Stocks



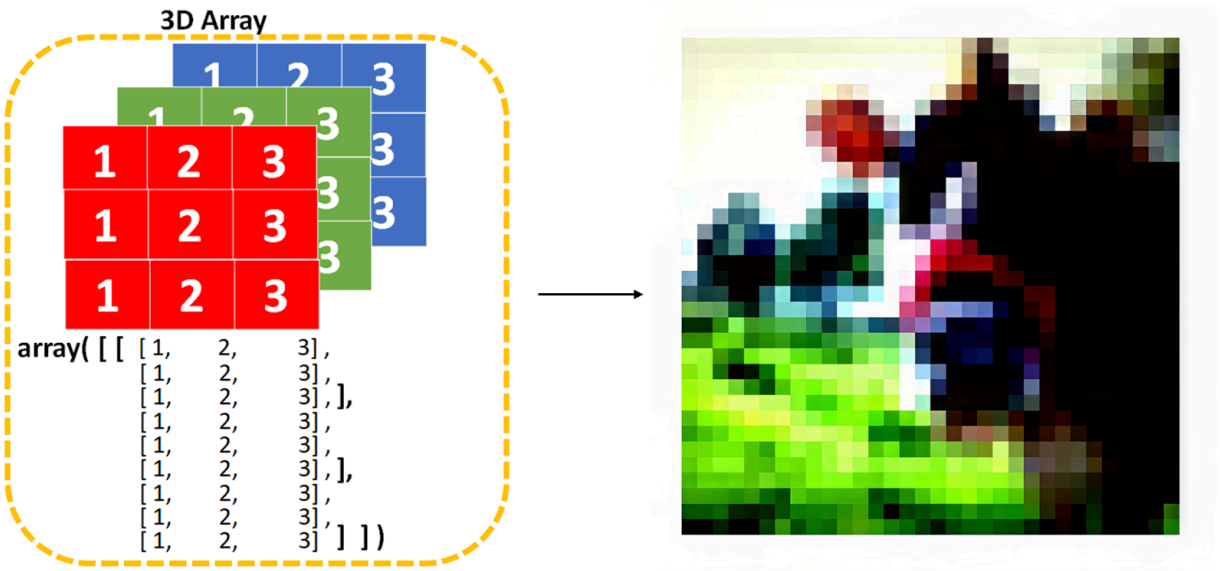
Example of Translating Stock Performance to 2D Images for Machine Learning

The study used four different datasets based on versions of the S&P 500 index, a well-known benchmark for the performance of the 500 largest publicly traded corporations in the United States. These datasets were standardized with the yfinance library, which contains historical market data from January 1, 1990, to January 1, 2024. The datasets included significant parameters such as open, close, adjusted close, volume, high, and low prices, which are important indications of stock performance and market trends.

However, in prior research, adding only these six variables resulted in little effectiveness with stock prediction. As a result, the generation of financial derivative data, also known as technical indicators, was utilized. This data extrapolates patterns that are performance-related. This technique increased the number of variables from six to 92.

Several data modifications were employed to compare the models' predictive power. One prominent transformation involved converting a 1-D array of stock values into a 2-D time series feature map. The basic dataset had 92 variables and 10 days of observation. This method identifies temporal links and trends throughout time, which are essential for developing accurate market predictions. Along with the regular S&P dataset, three extra variants were generated. This was accomplished by raising the depth of the 2-D time series to 100 days (Version 2), including US Treasury Bonds (Version 3), and S&P Futures (Version 4).

Images

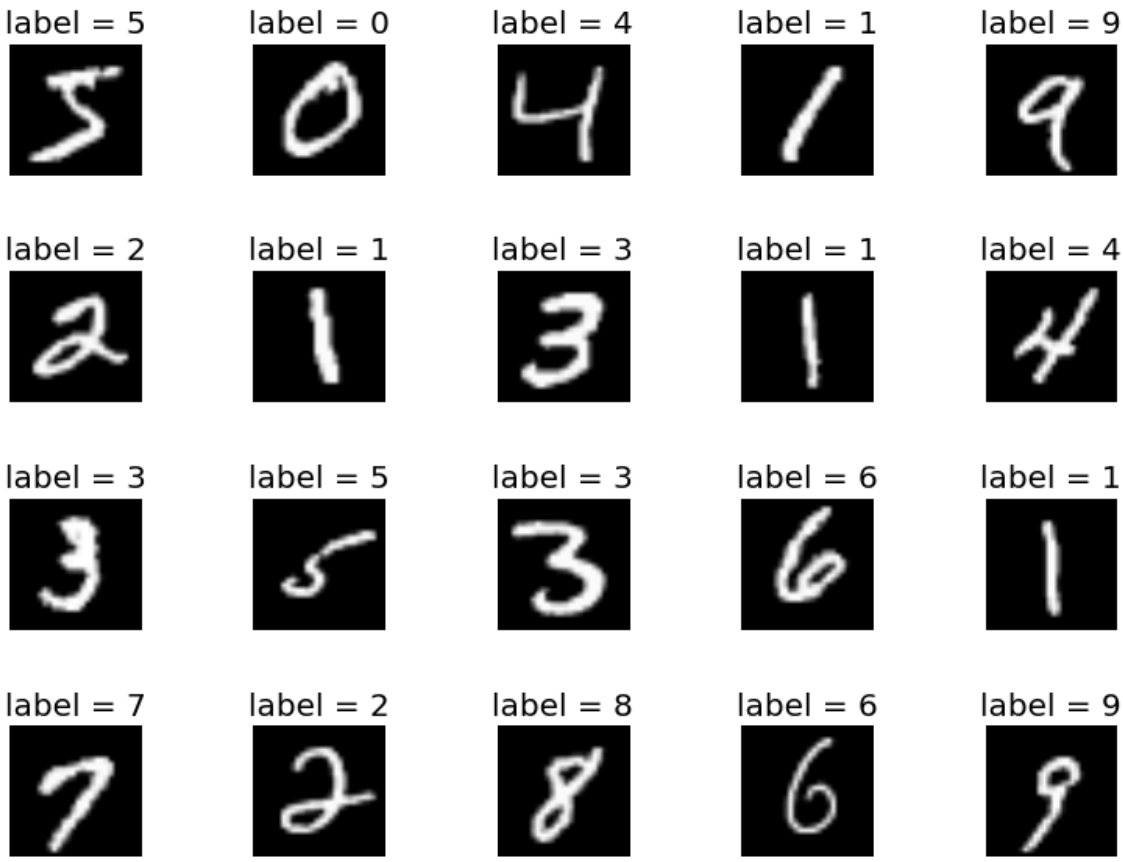


Example of the Dimensionality of the CIFAR Dataset: Images Consist of Height, Width, and Channels. Channels in this demonstration is the red, green, and blue colors in each pixel.

The study also used four well-known image datasets to evaluate the algorithms' performance on visual data. This data is displayed as a three-dimensional array of floating point

values, which includes height, width, and three channels (red, green, and blue). The goal of this dataset was to include a classification issue that used 3D data.

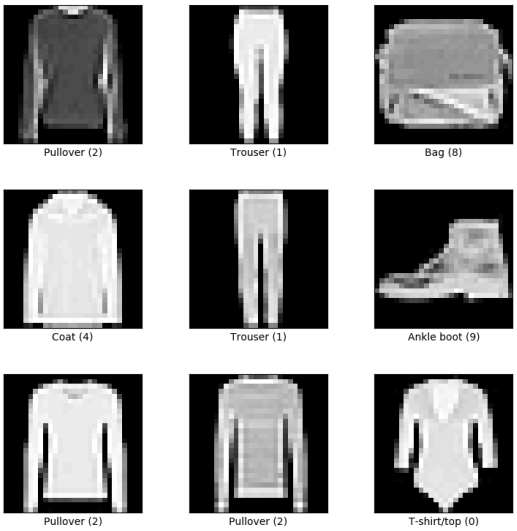
MNIST Dataset



Various MNIST Data Samples with Labels

The MNIST dataset contains 60,000 grayscale photographs of handwritten numbers (0–9). This dataset has served as a standard benchmark in machine learning for evaluating classification algorithm performance. Its simplicity and ease of use make it an excellent starting point for creating and testing image recognition models.

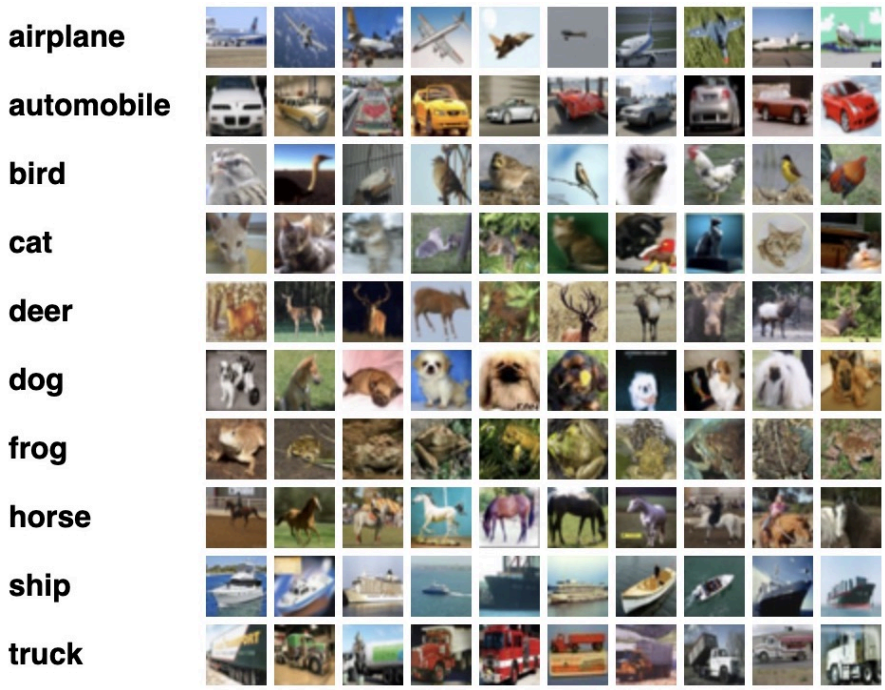
Fashion MNIST Dataset



Data Samples From the Fashion MNIST Project

Fashion MNIST, like the MNIST dataset, consists of 60,000 28×28 pixel images, but instead of digits, it includes 10 categories of fashion items like T-shirts, trousers, and shoes. This dataset presents a more difficult classification problem than MNIST, owing to the greater variety in the appearance of fashion items compared to handwritten numbers. Fashion MNIST provides a more appropriate baseline for modern picture classification tasks.

CIFAR-100



Various Data Points From CIFAR-100 Fine Dataset

The CIFAR-100 dataset, a subset of Tiny Images, includes 60,000 32×32 color images from 100 different classes. These classes are divided into 20 superclasses, each having fine (particular) and coarse (generic) labels, and both are employed in this study. Some excellent labels include maple, oak, palm, pine, and willow. However, these would all fall under the tree coarse category. Given that most people would struggle to spot these distinctions, this dataset presents a far more difficult task. The CIFAR-100 is frequently used to assess the performance of advanced image classification models and algorithms, especially deep learning techniques.

Summary of Datasets

Dataset Label	Problem Type	Outputs	Samples	1-D Features	2-D Features	3-D Features
CIFAR-100 Coarse Labels	Image Multi-Label Binary Classification	20 Labels	60,000 (50,000 Training, 10,000 Testing)	3072	(3,1024) [channels, features]	(3,32,32) [channels, x_features, y_features]
CIFAR-100 Fine Labels	Image Multi-Label Binary Classification	100 Labels	60,000 (50,000 Training, 10,000 Testing)	3072	(3,1024) [channels, features]	(3,32,32) [channels, x_features, y_features]
Fashion MNIST	Image Multi-Label Binary Classification	10 Labels	60,000 (50,000 Training, 10,000 Testing)	784	(1,784) [channels, features] and (28,28) [channels, features] were both tested	(1,28,28) [channels, x_features, y_features]
MNIST	Image Multi-Label Binary Classification	10 Labels	60,000 (50,000 Training, 10,000 Testing)	784	(1,784) [channels, features] and (28,28) [channels, features] were both tested	(1,28,28) [channels, x_features, y_features]
S&P 500 with 10 Days of Information	Multi-Output Binary Classification	10 Ouputs	5128 (1990 to 2024)	1012	(11, 92) [Sequence, Features]	N/A
S&P 500 with 100 Days of Information	Multi-Output Binary Classification	10 Ouputs	5128 (1990 to 2024)	9292	(101, 92) [Sequence, Features]	N/A
S&P 500 and US	Multi-Output Binary	10 Ouputs	5128 (1990 to 2024)	18584	(11, 184) [Sequence,	N/A

Treasury Bonds with 100 Days of Information	Classification				Features]	
S&P 500 and S&P 500 Futures with 100 Days of Information	Multi-Output Binary Classification	10 Ouputs	5128 (1990 to 2024)	18584	(101, 184) [Sequence, Features]	N/A

Graph Summarizing Datasets

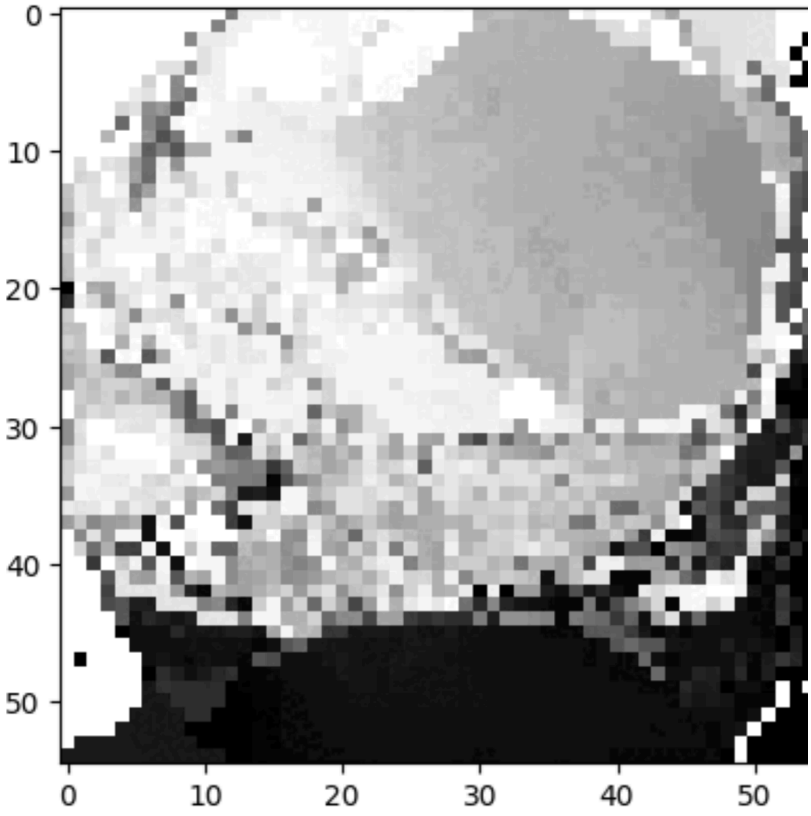
Dimensionality Augmentation

The datasets were divided into three categories: 1D, 2D, and 3D test sets. These were independently tested against their respective models: All machine learning approaches employed one-dimensional data. Because of their construction, LSTMs could only operate in two dimensions. To address this issue, the data was flattened to resemble a one-dimensional array: For example, transforming the shape [sequence length, features] into [1, sequence length x features] and feeding it to the LSTM while identifying it as 1-D. Convolutional Networks could use all three of these structures and were tested on them.

IGTD

Additionally, some resources were spent incorporating the Image Generator for Tabular Data (IGTD) algorithm. This was used to transform tabular data into images by assigning each feature to a unique pixel position, with similar features assigned to neighboring pixels and dissimilar features placed far apart. This transformation enables the use of CNNs in subsequent

analysis. However, due to the dominant CNN performance in early testing, this method was discarded to allow time for other models.



Sample from S&P500 Index with IGTD Processing: Notice how similar shaded pixels are grouped together. This is core principle of IGTD decomposition

Evaluating Performance

The Sci-Kit Learn Library was used to test models in this study. Performance was measured using four essential metrics: accuracy, precision, recall, and F1 score. Each indicatoroffers a unique view of the model's efficacy and data properties.

Accuracy

This metric calculates the fraction of correctly identified samples out of all samples. It provides a basic assessment of overall performance. However, accuracy alone might be

deceptive, particularly in unbalanced datasets when the quantity of samples in different classes changes dramatically. In such circumstances, forecasting the majority class may be sufficient to attain high accuracy.

Precision

Precision refers to the fraction of true positive predictions among all positive predictions made by the model. High precision indicates it is typically correct when the model predicts a positive class. This statistic is critical when the cost of false positives is large, such as medical diagnosis or fraud detection.

Recall

Recall, also known as sensitivity or true positive rate, measures how many true positive samples the model properly identifies. High recall is required when missing positive cases is extremely undesirable, such as disease screening or security threat detection.

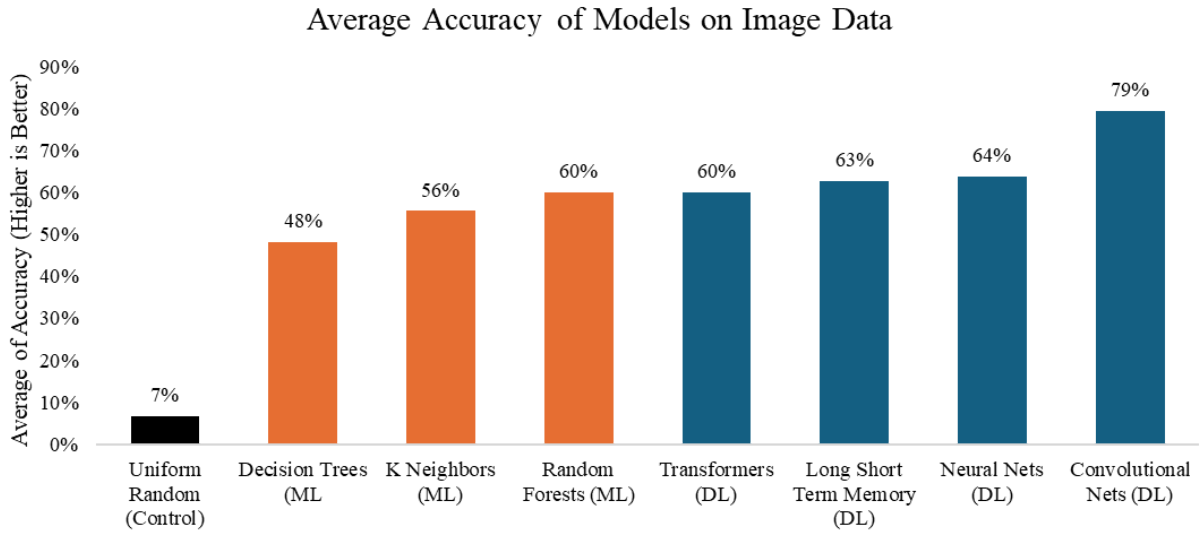
F1 Score

The F1 score is the harmonic mean of precision and recall, offering a single metric that balances both objectives. It is especially effective when the data has an unequal class distribution and both false positives and negatives are significant. The F1 score aids in understanding the trade-off between precision and recall, resulting in a more comprehensive evaluation of model performance. By incorporating these four indicators, the review process acquires a thorough grasp of the models' strengths and limitations, allowing for better decision-making and model refinement.

Results

The findings of this study highlight the distinct advantages of Deep Learning (DL) models over typical Machine Learning (ML) models, notably in the handling of complicated, multidimensional data.

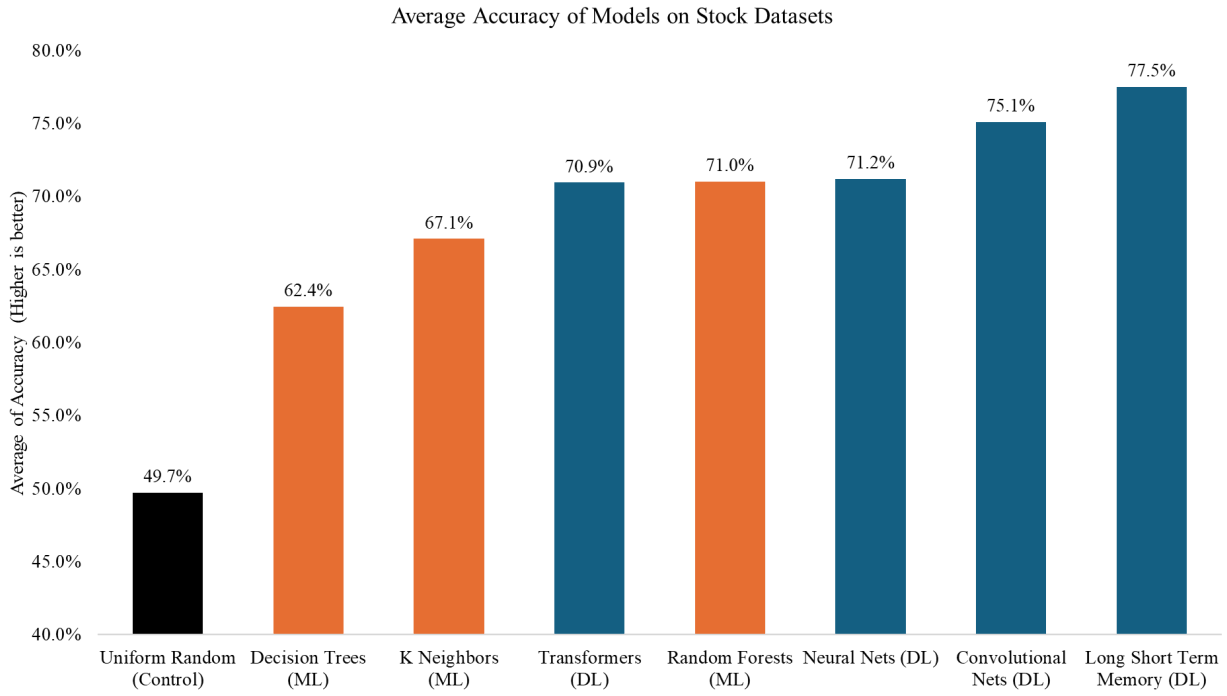
Image Datasets



Average Accuracy of Models on Image Data

DL models, like Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks, performed well on various datasets, particularly those containing images and sequential data. For example, CNNs acquired an average accuracy of 79% on picture datasets, much higher than the 60% attained by standard ML models such as Random Forests. This demonstrates DL models' ability to capture complicated patterns and spatial hierarchies in image data, making them perfect for image recognition and processing.

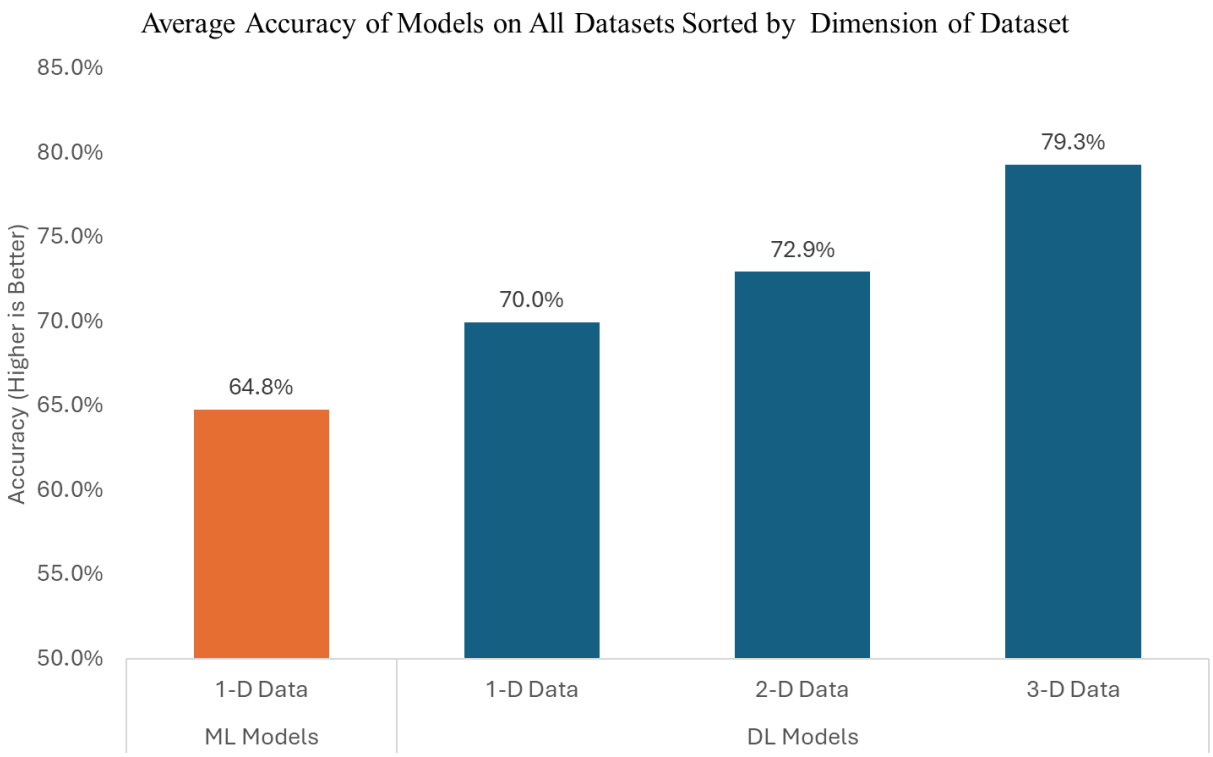
Stock Datasets



Average Accuracy of Models on Stpck Datasets

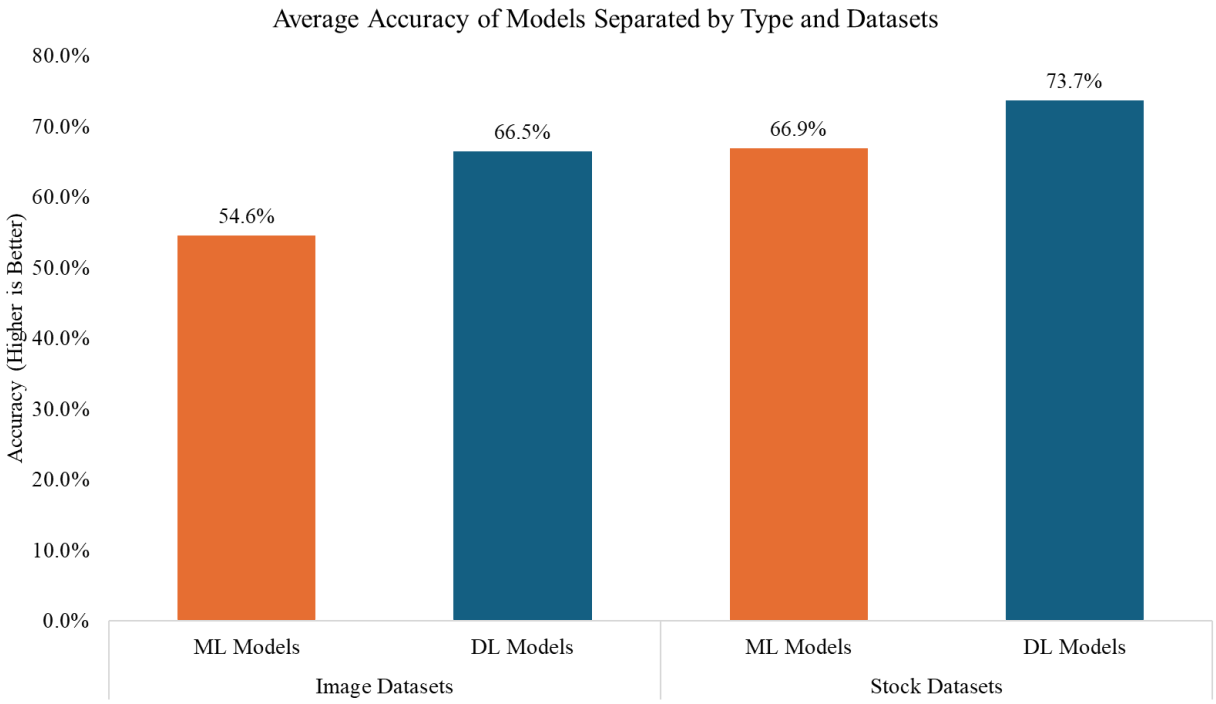
In stock data, DL models outperformed typical ML models, with LSTM networks obtaining an average accuracy of 77.5% versus 71% for Random Forests. This shows that DL models are more suited to deal with financial data's temporal relationships and sequential nature, resulting in more accurate predictions and insights. The capacity of DL models to analyze and learn from sequential data is critical in sectors like finance, where detecting trends and generating forecasts based on prior data is key.

Data Dimensionality



Average Accuracy of Models on All Datasets Sorted by Dimension of Dataset

When analyzing the impact of data dimensionality on model performance, DL models continuously improved as data dimensionality grew, with 3-D data having an average accuracy of 79.3%, compared to 72.9% for 2-D data and 70% for 1-D data. This shows that DL models may effectively use more data dimensions to improve their learning and prediction skills. Traditional ML models, on the other hand, did not show the same level of performance gains with greater data dimensionality, indicating a restriction in their ability to handle and use complicated data structures.



Average Accuracy of Models Seperated by Type and Datsets

When comparing results by model type (ML and DL) and dataset, it's evident that DL models generally outperform ML models. The performance advantage of DL models is especially pronounced in image datasets, showing a 22% improvement. In stock prediction, DL models still demonstrate a significant improvement of 10%, though it's less substantial than the improvement seen with images.

Analysis

In summary, the research highlights several critical insights into the performance of different ML and DL models:

DL models, particularly those utilizing complex architectures like CNNs and LSTMs, excel in handling multi-dimensional and sequential data. Their ability to capture intricate patterns and dependencies makes them suitable for complex tasks such as image recognition and time series forecasting.

Traditional ML methods, such as Random Forests, are sufficient for simpler problems with less complex data structures. These models are often easier to train and require less computational power, making them practical for many real-world applications.

Convolutional Neural Networks (CNNs) are versatile and effective across multiple datasets, especially those involving image data. Their design, which includes convolutional layers to detect spatial features, allows them to perform well in various image classification and recognition tasks.

Future Research

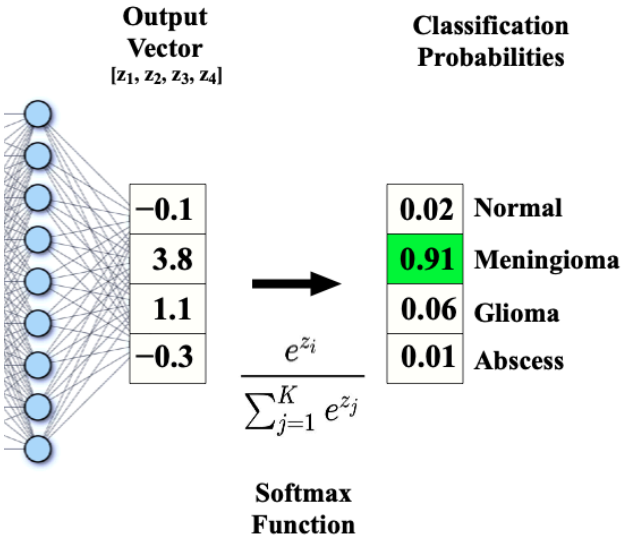
Future work aims to expand the scope of this benchmarking to include audio, video, and regression tasks. Each data type introduces new complexities and requires different model architectures and evaluation metrics. For example, audio and video data involve processing temporal sequences with additional spatial or auditory dimensions, often necessitating models like Long Short Term Memory (LSTMs) and transformers. Regression tasks, on the other hand, focus on predicting continuous values, where metrics like Mean Squared Error (MSE) are more appropriate.

Pitfall: Challenges in Creating a Unified Index

Developing a uniform performance index is difficult since different machine learning workloads and models necessitate unique evaluation criteria. Classification tasks often employ metrics like accuracy, precision, recall, and F1 score to evaluate the model's ability to categorize data items accurately. In contrast, regression tasks use metrics like Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) to calculate the difference between predicted and actual values. These disparities make it difficult to compare performance across tasks without using a well-defined composite statistic.

Two potential options come to mind. Two model performance indexes might be created: one for classification (using accuracy as a metric) and one for regression (using MSE). This method is most appealing since it tackles the two problem kinds separately and best illustrates the models' capabilities.

Another possible method is to combine an index with a universal evaluation metric, such as MSE, that can be applied to all issue types. This system's potential stems from its single, universal index, which other scholars may easily reference. However, the main challenge with this technique is that MSE may only be suitable for some problem types. For example, image classification tasks use the Soft-Max function to determine whether a label is valid. Even if Soft-Max selects the correct value 100% of the time, the mean square error increases if the incorrect labels are partially activated, resulting in a worse evaluation.



Example of Softmax Function: the values for the incorrect labels would lower performance with MSE evaluation despite the correct label being selected.

Overall, this research provides valuable insights into the strengths and limitations of various ML and DL models, guiding future work and practical applications in the field. Despite

its challenges, the development of a unified performance index represents a significant step towards a more comprehensive and standardized model evaluation.

References

Agrawal, A. (2020, August 18). *How to Win with Machine Learning*. Harvard Business Review. https://hbr.org/2020/09/how-to-win-with-machine-learning

CS231N Convolutional Neural Networks for Visual Recognition. (n.d.). https://cs231n.github.io/convolutional-networks/

Krizhevsky, A. (2009). *Learning Multiple Layers of Features from Tiny Images*. https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf

Krizhevsky, A., Nair, V., & Hinton, G. (n.d.). *CIFAR-10 and CIFAR-100 datasets*. Retrieved June 7, 2024, from https://www.cs.toronto.edu/~kriz/cifar.html

LeCun, Y., Cortes, C., & Burges, C. (n.d.). *MNIST handwritten digit database*. Yann Lecun. Retrieved June 7, 2024, from http://yann.lecun.com/exdb/mnist/

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017, June 12). *Attention is all you need*. arXiv.org. https://arxiv.org/abs/1706.03762

Xiao, H., Rasul, K., & Vollgraf, R. (2017, August 25). *Fashion-MNIST: A MNIST-like fashion product database*. GitHub. Retrieved June 7, 2024, from https://github.com/zalandoresearch/fashion-mnist

Yahoo. (n.d.-a). *Interest rate 10 year Treasury notes*. Yahoo Finances. Retrieved June 7, 2024, from https://finance.yahoo.com/quote/%5ETNX/

Yahoo. (n.d.-b). *S&P 500 Futures Index*. Yahoo Finances. Retrieved June 7, 2024, from https://finance.yahoo.com/quote/ES%3DF/

Yahoo. (n.d.-c). *S&P 500 Index*. Yahoo Finances. Retrieved June 7, 2024, from https://finance.yahoo.com/quote/%5EGSPC/

Zhu, Y., Brettin, T., Xia, F., Partin, A., Shukla, M., Yoo, H., Evrard, Y. A., Doroshow, J. H., & Stevens, R. L. (2021). Converting tabular data into images for deep learning with convolutional neural networks. *Scientific Reports*, *11*(1). https://doi.org/10.1038/s41598-021-90923-y