Examining Neural Networks Through Architectural Variation Analysis for Image Classification

HAIXIA LIU*, TIM BRAILSFORD, and LARRY BULL, The School of Computing and Creative Technologies, University of the West of England, UK

JAMES GOULDING and GAVIN SMITH, N/Lab, The University of Nottingham, UK

This paper presents a method for examining neural networks in image classification through architectural variation analysis. Small-scale experiments generate initial insights, and the configurations are further tested on entire datasets. The newly proposed sampling strategy, which focuses on heavily confused samples to identify instance hardness, offers researchers a way to explore novel methods with reduced computational costs. Image pre-processing operations are crucial in the image classification pipeline. Applying image sharpening prior to other standard pre-processing techniques was found to yield improved results. The choice and order of layers significantly impact model performance. We propose three layer-level operations: Plug and Play (PaP), Leave One Layer Out (LOLO), and Select and Reorder (SaRe). The results indicate that convolutional (Conv2D) and batch normalization (BN) layers are significant in image classification tasks, but this is dependent upon the context of the images. Performing BN before Conv2D can improve the model's predictive capability. This study provides valuable insights into optimizing deep learning models, with potential avenues for future research, including explainable AI (XAI).

ACM Reference Format:

Haixia Liu, Tim Brailsford, Larry Bull, James Goulding, and Gavin Smith. 2024. Examining Neural Networks Through Architectural Variation Analysis for Image Classification. 1, 1 (June 2024), 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

CCS CONCEPTS

Computing methodologies • Machine learning algorithms

KEYWORDS

Architectural Variation Analysis, Explainable AI, Layer Selection, Layer Ordering, Deep Learning Optimization, Neural Networks, LOLO

1 INTRODUCTION

Deep learning applications have gained significant importance in recent years, with widespread use in areas ranging from Generative Pre-trained Transformers (GPT) to Residual Networks (ResNet). However, these algorithms are often viewed as "black boxes", making it difficult for individuals, especially those without expertise in AI techniques and terminology, to understand the reasoning behind their results. This has spurred growing interest in Explainable AI

*Corresponding author Email: haixia.liu@uwe.ac.uk

Authors' addresses: Haixia Liu; Tim Brailsford; Larry Bull, The School of Computing and Creative Technologies, University of the West of England, Coldharbour Lane, Bristol, UK, BS16 1QY; James Goulding; Gavin Smith, N/Lab, The University of Nottingham, Jubilee Campus, Nottingham, UK, NG8 1BB.

Manuscript submitted to ACM

(XAI), an approach that seeks to clarify deep networks' processes and representations by frequently integrating both transparent and black box models.

Modifications to deep network architectures often involve adjusting individual layer specifics or reordering layer combinations. For instance, in image classification, deep learning networks typically consist of Batch Normalization (BN) layers, Convolutional Layers (Conv2D), Pooling Layers (PL), Dropout Layers (DropL), Fully Connected Layers (FCL), and Activation Layers (AL). Altering the details of these layers or changing their combinations can not only improve model performance but also deepen understanding of the impact of the architecture. XAI researchers work to better understand how these modifications enable the model to arrive at its predictions.

In this paper, we demonstrate that even minor adjustments to existing models can substantially impact their performance. We employ small-scale experiments to generate initial 'inspirations' which can then be used to validate larger datasets and refine existing algorithms. Furthermore, we examine the effects of dataset characteristic differences on model performances. By focusing on the specific changes that lead to improved outcomes for various data collections, we are aiming to optimise image classification and gain a better insight into how models function with different datasets.

2 SCOPE, RELATED WORK AND CONTRIBUTION

In this study, we have focused entirely on image classification. Similar issues face other problems, such as natural language processing (NLP) or U-Net, but the results may well differ and we are making no claims about generalisation. The motivation behind this work arises from the observation that powerful models are almost always developed incrementally over time, and the process of discovering these models tends to be expensive. The fundamental elements of these models are mathematical concepts and algorithms, thus it is likely that recombining and reordering these components could lead to new discoveries. It is, though, important to recognise that incorporating novel components can often also have significant impact.

Our explorations are divided into three aspects: sampling strategy; image processing and neural network layers. We challenge the traditional practices of ordering operations (e.g. image enhancements and network layers), by changing those orders and examining the impact this has upon performance.

2.1 Sampling strategy

In order to reduce the computational cost of both supervised and unsupervised learning algorithms, it is usually necessary to scale down the dataset using one of a variety of techniques to select a sample of data that is representative of the entire dataset [5]. Stratified-statistical sampling methods have usually been found to generate the highest classification accuracy [13].

More recent studies have explored the use of sampling strategies for both hard and easy problems. It has been found that under some circumstances, at least, that machine-learning-assisted Monte Carlo approaches fail to correctly sample computationally hard problems [3]. In large datasets class imbalance is potentially a major problem that has been addressed by utilising heuristic sampling methods such as SMOTE (Synthetic Minority Oversampling Technique), in combination with cleaning strategies [14]. SMOTE has in recent years become the *de facto* standard method of handling unbalanced datasets [6]. However, unbalanced datasets remain a challenge in many real-world scenarios. It has been proposed that taking instance hardness into account be a useful way of improving performance with such datasets [2].

We have explored a sampling strategy that utilises instance hardness as determined by the confusion matrix generated by simple machine learning algorithms (such as the k-NN classifier for the MNIST dataset [9], Histogram of Oriented Gradients (HOG) feature descriptor using SVM for the FMNIST dataset [8] and logistic regression for the CIFAR-10

dataset [12]). We then use the confusion matrices generated by these algorithms to extract the samples that are heavily confused with each other. If one algorithm beats another on classifying heavily confused samples, then we would expect that algorithm to be stronger than others for classifying other samples.

It is important to note that the purpose of our sampling strategy is to compare the relative performance of novel methods. We aim to answer the question that if model1 outperforms model2 using the hardALL subset, then can we safely assume that model1 will likewise outperform model2 using the entire dataset. This is important, because if that is indeed the case then researchers studying a novel models efficiency could carry out preliminary experiments using only the HardALL set to save time and potentially significant computational cost.

2.2 Image Pre-processing

Image pre-processing consists of a series of operations intended to improve the image data by suppressing distortion and enhancing important features that are useful in classification [15]. This involves converting them from the RGB to BGR colour space (for framework and hardware compatibility purposes) and then zero cantering each colour channel with respect to the ImageNet dataset [4]. This involves adjusting the pixel values of each colour channel in the image such that the mean becomes zero, based on the mean values derived from the entire dataset, without changing the original range of the pixel values [10]. Image enhancements, such as sharpening, are often applied after pre-processing in order to make images more distinct and yield better results [7]. In this study, we reordered the steps of image pre-processing and enhancement to amplify the salient features by sharpening the images before applying other standard pre-processing techniques.

2.3 Neural Network Layers

Adding some layers from a baseline sequence of layers in a typical neural network architecture also has the potential to be useful. For example, [16] used a few lightweight adapters, which were added to the existing image model, and they found that these adapters can help achieve comparable or even better performance. Inspired by this, we explored layer level operation using plug and play **(PaP)** to examine the single layer efficacy. Leave One Feature Out (LOFO) is an algorithm that estimates the importance of a feature by iteratively removing each one from the set, and then evaluating the performance of the model[11]. Inspired by LOFO, we devised an algorithm to explore the effect of cutting out individual layers. We have called this Leave One Layer Out **(LOLO)**.

The optimal placement of Batch Normalization (BN) within Convolutional Neural Networks (CNNs) has been studied by [10]. They recommend inserting the BN transform immediately before the nonlinearity in each layer, which indicates that the BN layer should be inserted between the convolutional layer and the activation function. This would mean performing BN after Conv2D. In this study, we explored the impact of swapping the BN and Conv2D layers and we also went further to explore Select-and-Reorder **(SaRe)** operations on other layers than BN and Conv2D.

We have demonstrated a need for a framework that will facilitate this by providing a standardised sequence of operations for such comparisons, and we have outlined what this framework needs to consist of. The framework we propose (see Fig. 1) is inspired by Brunner's discovery learning principles [1], which is the foundation of constructivist theories of learning. The framework aims to enhance: critical thinking; creative problem-solving; meaningfulness; exploration and collaborative feedback. The framework we are proposing involves selecting an architecture based on prior knowledge, exploring various architectural permutations, and then analysing and comparing the results at each stage.

2.4 Research questions

The research questions of this study are as follows:

- To what extent is our Easy/Hard sampling strategy useful in evaluating different models in comparison with experimenting on the entire dataset? Do we observe different behaviors across different image sets (MNIST, FMNIST, CIFAR-10)?
- Can sharpening, when performed before other image pre-processing steps, improve the model's classification performance?
- **PaP**: When a single layer is plugged in, which one shows the best performance? Is this consistent across all datasets?
- **LOLO**: Removing which layer results in the most significant degradation of the existing model? Is this consistent across all datasets?
- **SaRe**: Swapping which layers has the greatest effect on the model's performance? Is this consistent across all datasets?

3 METHODOLOGY

3.1 Datasets

The experiments were implemented in Python using Keras – with the following data sets: CIFAR-10, Fashion MNIST (FMNIST) [1], MNIST [2] and MedMNIST [3]. Table 1 displays the resulting subset characteristics, and sample size denotes the sum of training and validation samples.

We started by experimenting using small datasets (HardALL and EasyALL), comparing the results generated by each model. Where results were consistent and interesting observations were made, we repeated the experiments with larger datasets.

3.2 Pre-processing

Data was preprocessed using the Tensorflow preprocess_input function [4]. The CIFAR10 images were originally sized 32 by 32. Inorder to explore upsampling they were resized to 256 by 256 before being converted to numpy arrays. The image resizing used UpSampling2D layer from Keras [5], utilizing nearest neighbour interpolation. We tried doing the image enhancement both before and after the resizing.

3.3 Architectural Variations

Following pre-processing, various layer operations were performed for comparative purposes. Plug-and-Play (PaP) was used to evaluate the impact that adding single layers have on the performance of the model using different datasets. The "Leave One Layer Out" (LOLO) technique was used to evaluate the contribution or importance of each individual layer. This method involves systematically removing one layer at a time from the network, and observing the impact on performance. We also used Select-and-Reorder (SaRe) to investigate the impact of two or more layers' placement on the network.

[1] tf.keras.datasets
[2] sklearn.datasets
[3] https://medmnist.com
[4] https://www.tensorflow.org/api_docs/python/tf/keras/applications/resnet50/preprocess_input
[5] https://keras.io/api/layers/reshaping_layers/up_sampling2d/

Dataset	Class names (sample size)	Image size and color
MNIST-HardALL	4(6824) and 9(6958)	28by28, grayscale
MNIST-EasyALL	1(7877) and 5(6313)	28by28, grayscale
MNIST-ALL	(70,000)	28by28, grayscale
FMNIST-HardALL	T-shirt/top(7000) and Shirt(7000)	28by28, grayscale
FMNIST-EasyALL	Ankle boot(7000) and Bag(7000)	28by28, grayscale
FMNIST-ALL	(70,000)	28by28, grayscale
CIFAR-10-HardALL	dog (6000) and cat (6000)	32by32, 3channel
CIFAR-10-EasyALL	car (6000) and deer (6000)	32by32, 3channel
CIFAR-10-ALL	(60000)	32by32, 3channel
PneumoniaMNIST-ALL	(5,856)	28by28, grayscale
TissueMNIST-ALL	(236,386)	28by28, grayscale
PathMNIST-ALL	(107,180)	28by28, 3channel
BloodMNIST-ALL	(17,092)	28by28, 3channel
OrganAMNIST-ALL	(58,850)	28by28, grayscale
OrganCMNIST-ALL	(23,660)	28by28, grayscale
OrganSMNIST-ALL	(25,221)	28by28, grayscale
BreastMNIST-ALL	(780)	28by28, grayscale
DermaMNIST-ALL	(10,015)	28by28, 3channel
OCTMNIST-ALL	(109,309)	28by28, grayscale

Table 1. Datasets for exploration using full data or different classes with (sample sizes).

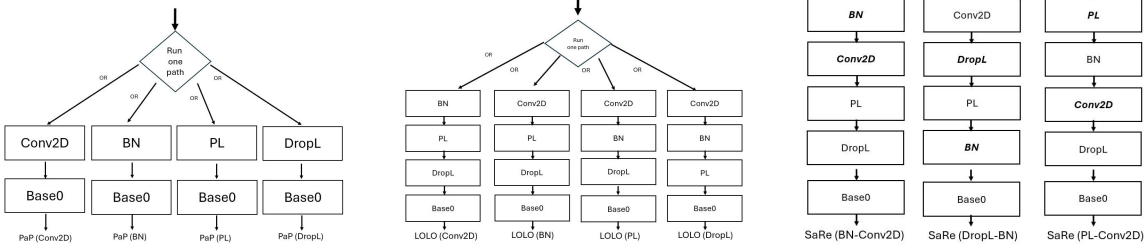
In additional to these layer operations, we also experimented using DropL to disconnect some individual neurons and Batch Normalisation (BN) – evaluating the impact that each of these permutations had upon the performance of the network.

We used different base models to study the network performance: Base0 was a simple neural network with fully connected layer (FCL) and Activation layer (AL), which is used for exploring PaP. BaseSeq consisted of Conv2D-BN-PL-DropL-FCL-AL (see Table 2), which is used for exploring LOLO and SaRe.

Abbreviation	Layer Name
Conv2D	Convolutional Layer
BN	Batch Normalisation
PL	Pooling Layer
DropL	Dropout Layer
FCL	Fully Connected Layer
AL	Activation Layer

Table 2. Ordering of the Baseseq layers.

Fig. 1 illustrates our architectural variations framework.



(a) Plug and Play (PaP) operation examining each individual layer by adding only one layer on top of Base0.

(b) Leave One Layer Out (LOLO) is based on the Baseseq as shown on Table 2.

(c) Select and Reorder (SaRe) is based on the original Baseseq shown on Table 2. The reordered layers are shown in bold.

Fig. 1. Flow charts of layer level operations of the framework. The results of each operation on different datasets are shown in Fig. 2a, 3a and 4a.

After completing these small scale experiments we conducted larger experiments using all samples from MNIST, FashionMNIST and CIFAR-10. Finally we tested our findings by repeating the process using the models that had been identified as interesting on different datasets (MedMNIST).

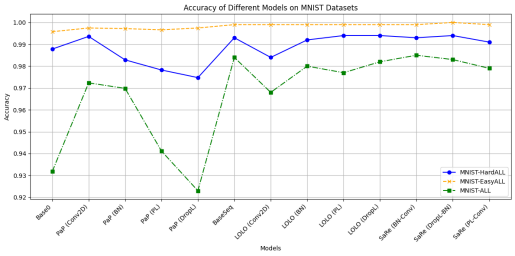
3.4 Experimental Settings

The parameters used for all of the experiments were as follows: testsize=0.25, randstate=42, batchsize=16, number of epochs=100, validationsplit=0.25, dropoutrate=0.25. All reported accuracy scores are based on the validation sets.

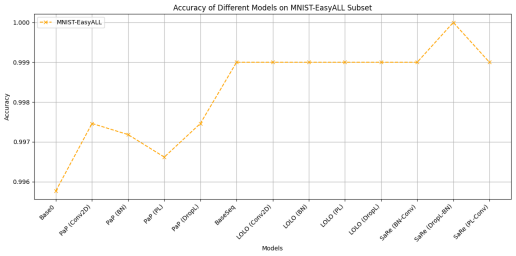
4 RESULTS

The results of different models on different datasets are shown in Fig. 2a 3a and 4a. Comparing HardALL with ALL, we can see that where models perform poorly on HardALL then they also perform poorly on ALL. Essentially the dips in performance are amplified with ALL.

Different datasets exhibit distinct characteristics. For FMNIST, ten out of thirteen models performed better on the entire dataset compared to the HardALL subset. This contrasts with the trends seen in the MNIST and CIFAR-10 datasets, where all models performed better on the HardALL subset than on the entire dataset.

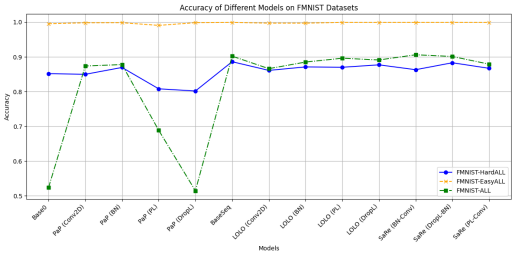


(a) Accuracy of each model on the MNIST dataset.

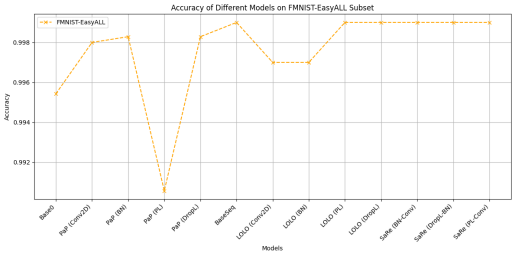


(b) Accuracy of each model on the MNIST-EasyALL subset, scaled to make the patterns more obvious.

Fig. 2. Accuracy of each model on the MNIST-EasyALL dataset.

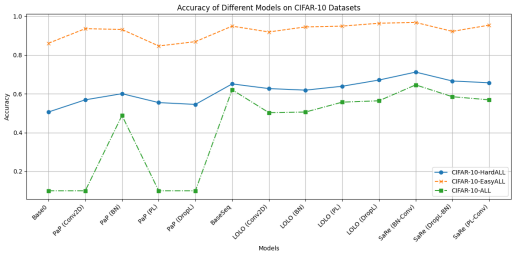


(a) Accuracy of each model on the FMNIST dataset.

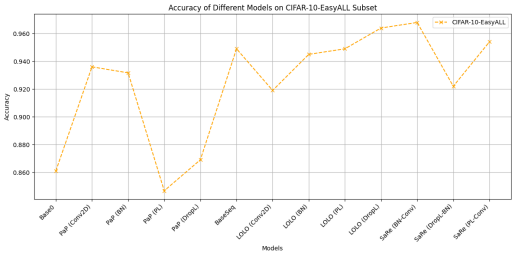


(b) Accuracy of each model on the FMNIST-EasyAll subset, scaled to make the patterns more obvious.

Fig. 3. Accuracy of each model on the FMNIST dataset.



(a) Accuracy of each model on the CIFAR-10 dataset.



(b) Accuracy of each model on the CIFAR-10-EasyAll subset, scaled to make the patterns more obvious.

Fig. 4. Accuracy of each model on the CIFAR-10 dataset.

Detailed results with accuracy numbers are shown in Tables 3, 4 and 5. We considered the best performing model, excluding any ties. PaP (Conv2D) and PaP (BN) were the best 3 and 4 times respectively, while SaRe (BN-Conv) was the best 5 times and SaRe (DropL-BN) 3 times. Based on Table 4, removing the Conv2D layer consistently leads to the most significant performance degradation across most datasets. This highlights the importance of the Conv2D layer in maintaining high model accuracy. However, for CIFAR-10-HardALL subset, the removal of the Batch Normalization

(BN) layer resulted in worse performance, with accuracy decreasing from 0.651 to 0.619, indicating the critical role of BN in maintaining model accuracy for this subset. Based on Table 5, SaRe (BN-Conv) emerges as the most successful configuration, winning 5 times, followed by SaRe (DropL-BN), which prevails 3 times.

Dataset name	Base0	PaP (Conv2D)	PaP (BN)	PaP (PL)	PaP (DropL)
MNIST-HardALL	0.988	**0.994**	0.983	0.978	0.975
MNIST-EasyALL	0.996	0.997	0.997	0.997	0.997
MNIST-ALL	0.932	**0.972**	0.970	0.941	0.923
FMNIST-HardALL	0.852	0.850	**0.869**	0.808	0.802
FMNIST-EasyALL	0.995	0.998	0.998	0.991	0.998
FMNIST-ALL	0.524	0.874	**0.878**	0.689	0.515
CIFAR-10-HardALL	0.506	0.569	**0.601**	0.555	0.545
CIFAR-10-EasyALL	0.861	**0.936**	0.932	0.847	0.869
CIFAR-10-ALL	0.100	0.100	**0.488**	0.100	0.100

Table 3. PaP Results of MNIST, FMNIST and CIFAR-10 datasets. Bold indicates the best result of each.

Dataset name	BaseSeq	LOLO (Conv2D)	LOLO (BN)	LOLO (PL)	LOLO (DropL)
MNIST-HardALL	0.993	**0.984**	0.992	0.994	0.994
MNIST-EasyALL	0.999	0.999	0.999	0.999	0.999
MNIST-ALL	0.984	**0.968**	0.980	0.977	0.982
FMNIST-HardALL	0.886	**0.861**	0.871	0.870	0.877
FMNIST-EasyALL	0.999	0.997	0.997	0.999	0.999
FMNIST-ALL	0.902	**0.866**	0.885	0.896	0.891
CIFAR-10-HardALL	0.651	0.627	**0.619**	0.639	0.671
CIFAR-10-EasyALL	0.949	**0.919**	0.945	0.949	0.964
CIFAR-10-ALL	0.621	**0.503**	0.506	0.557	0.564

Table 4. LOLO Results of MNIST, FMNIST and CIFAR-10 datasets. The worst result of each is shown in bold indicating the significance of the layer.

Dataset name	SaRe (BN-Conv2D)	SaRe (DropL-BN)	SaRe (PL-Conv2D)
MNIST-HardALL	0.993	**0.994**	0.991
MNIST-EasyALL	0.999	**1.000**	0.999
MNIST-ALL	**0.985**	0.983	0.979
FMNIST-HardALL	0.863	**0.883**	0.867
FMNIST-EasyALL	0.999	0.999	0.999
FMNIST-ALL	**0.906**	0.901	0.879
CIFAR-10-HardALL	**0.712**	0.666	0.657
CIFAR-10-EasyALL	**0.968**	0.922	0.954
CIFAR-10-ALL	**0.646**	0.585	0.569

Table 5. SaRe Results of MNIST, FMNIST and CIFAR-10 datasets. Bold indicates the best result of each.

Using PaP for all data sets, we found that HardALL subsets provides a better indication of the relative performance of the models than EastALL. Hence using the EasyALL subsets could be misleading and the focus for exploration should

be on the more challenging examples. Using LOLO, similar patterns are observed for MNIST and FMNIST, but not for CIFAR-10. This may well be due to the far more diverse image in CIFAR-10 than in the other data sets although then there is very little to choose between Conv2D and BN. Using SaRe, we found that swapping the order of BN and Conv2D always produces better results. Swapping DropL and BN produces better results for the less diverse images in MNIST and FMNIST, but not for CIFAR-10. Swapping PL and Conv2D never makes any difference. To summarise this, the context of the images matters when optimising the architecture of networks. There are great performance gains to be made, but only for certain types of images.

The studies using MNIST, FMNIST and CIFAR-10 have been repeated for the most promising architectural variations including PaP(Conv2D), PaP(BN), Baseseq and SaRe(BN-Conv2D), using the MedMNIST data set which is a large collection of real-world biomedical images. The results are shown in Table 6. The best performing variations are SaRe(BN-Conv2D) (7 out of 10) and PaP(Conv2D) (3 out of 10).

Dataset name	**Best Model**
PneumoniaMNIST-ALL	SaRe (BN-Conv2D)
TissueMNIST-ALL	SaRe (BN-Conv2D)
PathMNIST-ALL	SaRe (BN-Conv2D)
BloodMNIST-ALL	PaP (Conv2D)
OrganAMNIST-ALL	SaRe (BN-Conv2D)
OrganCMNIST-ALL	SaRe (BN-Conv2D)
OrganSMNIST-ALL	SaRe (BN-Conv2D)
BreastMNIST-ALL	PaP (Conv2D)
DermaMNIST-ALL	PaP (Conv2D)
OCTMNIST-ALL	SaRe (BN-Conv2D)

Table 6. Best performing models using MedMNIST.

By comparing the impact upon performance that different ordering of transformations has using the CIFAR-10-HardALL dataset, we conclude that sharpening followed by input pre-processing is better than the reverse and the use of UpSampling2D improves performance.

5 DISCUSSION

In this study, we have conducted an architectural variation analysis of neural networks used for image classification. Our goal is to provide an approach that assists researchers in testing novel methods on smaller datasets, which could provide a valuable shortcut when computational resources are limited. The rationale is if model adjustments impact performance on smaller datasets, then it is likely to be worth conducting further experiments on larger datasets to validate those findings. We have used a variety of technique to explore the impact of architectural variation (PaP, LOLO and SaRe) using several standard datasets (MNIST, FMNIST and CIFAR-10) taking instance hardness into account. We then attempted to validate our findings using the larger real-world MedMNIST dataset. We have found that although there is great promise in this approach, it does depend upon the context of the images. We found that in various ways the real-world images of CIFAR-10 behave differently to the simplified images of MNIST and FMNIST. In the future we hope to use more detailed dataset characteristics alongside statistical tests to better elucidate the observed trends and findings. It should be possible to use mathematical approaches to derive an understandable prototype and enhance model explainability through a comprehensive exploration framework.

REFERENCES

[1] Jerome S Bruner. 1961. The act of discovery. *Harvard educational review* (1961).

[2] Angelos Chatzimparmpas, Fernando Vieira Paulovich, and Andreas Kerren. 2023. Hardvis: Visual analytics to handle instance hardness using undersampling and oversampling techniques. In *Computer Graphics Forum*, Vol. 42. Wiley Online Library, 135–154.

[3] Simone Ciarella, Jeanne Trinquier, Martin Weigt, and Francesco Zamponi. 2023. Machine-learning-assisted Monte Carlo fails at sampling computationally hard problems. *Machine Learning: Science and Technology* 4, 1 (2023), 010501.

[4] Jia Deng. 2009. A large-scale hierarchical image database. *Proc. of IEEE Computer Vision and Pattern Recognition, 2009* (2009).

[5] Amr ElRafey and Janusz Wojtusiak. 2017. Recent advances in scaling-down sampling methods in machine learning. *Wiley Interdisciplinary Reviews: Computational Statistics* 9, 6 (2017), e1414.

[6] Alberto Fernández, Salvador Garcia, Francisco Herrera, and Nitesh V Chawla. 2018. SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. *Journal of artificial intelligence research* 61 (2018), 863–905.

[7] Agam Das Goswami. 2023. Automatic classification of the severity of knee osteoarthritis using enhanced image sharpening and cnn. *Applied Sciences* 13, 3 (2023), 1658.

[8] KV Greeshma and K Sreekumar. 2019. Fashion-MNIST classification based on HOG feature descriptor using SVM. *International Journal of Innovative Technology and Exploring Engineering* 8, 5 (2019), 960–962.

[9] Divas Grover and Behrad Toghi. 2020. MNIST dataset classification utilizing k-NN classifier with modified sliding-window metric. In *Advances in Computer Vision: Proceedings of the 2019 Computer Vision Conference (CVC), Volume 2 1*. Springer, 583–591.

[10] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*. pmlr, 448–456.

[11] SM Mohidul Islam and Farhana Tazmim Pinki. 2018. Partial View-Based Object Recognition using Leave-one-out Approach with Classification and Regression Trees. *Journal of Computer Technology and Applications* 9 (2018), 9–16.

[12] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).

[13] Christopher A Ramezan, Timothy A Warner, and Aaron E Maxwell. 2019. Evaluation of sampling and cross-validation tuning strategies for regional-scale machine learning classification. *Remote Sensing* 11, 2 (2019), 185.

[14] Erendira Rendon, Roberto Alejo, Carlos Castorena, Frank J Isidro-Ortega, and Everardo E Granda-Gutierrez. 2020. Data sampling methods to deal with the big data multi-class imbalance problem. *Applied Sciences* 10, 4 (2020), 1276.

[15] Milan Sonka, Vaclav Hlavac, and Roger Boyle. 1993. Image pre-processing. *Image processing, analysis and machine vision* (1993), 56–111.

[16] Taojiannan Yang, Yi Zhu, Yusheng Xie, Aston Zhang, Chen Chen, and Mu Li. 2023. AIM: Adapting Image Models for Efficient Video Understanding. In *International Conference on Learning Representations*. https://openreview.net/forum?id=CIoSZ_HKHS7