# The Investigation of Multiple Optimization Methods on Convolutional Neural Network

Hetian Pan *

Department of Data Science and Big Data Technology, The Chinese University of Hong Kong, Shenzhen, China

* Corresponding author: hetianpan@link.cuhk.edu.cn

**Abstract.** In this study, the optimization of a Convolutional Neural Network (CNN) was conducted using the Fruits 360 dataset, with a specific emphasis on the impacts of Spatial Transformer Network (STN) and Stochastic Gradient Descent (SGD) optimization methods. Firstly, a baseline CNN model is built, which achieves 97.84% accuracy with a loss of 0.0999 after 50 epochs. Then, the impact of integrating STN and SGD into CNN models separately is investigated. The addition of STN slightly increased the accuracy to 97.92%, reduced the loss to 0.0994, and decreased the validation accuracy. This result suggests that while STN enhances the model's generalization ability, it may slightly reduce the maximum accuracy achievable on the validation set. After SGD optimization, the verification accuracy is increased to 98.19%, the loss is reduced to 0.0537, and the verification accuracy is increased to 98.40%. These results highlight the effectiveness of SGD in fine-tuning model parameters, resulting in more accurate models and improved generalization capabilities. A comparative analysis of these methods highlights their respective advantages. The effectiveness of the STN is rooted in its capacity to improve model generalization and mitigate overfitting, which is particularly beneficial in situations that demand robustness against varied data sets. In contrast, SGD stands out for its ability to significantly improve model accuracy and reduce loss, making it a balanced choice for comprehensive model optimization. Future research directions include exploring these optimization techniques on various datasets and investigating the potential of combining STN and SGD to achieve higher performance in CNN models.

**Keywords:** CNN, SGD, STN.

## 1. Introduction

In recent times, the advent of deep learning along with convolutional neural networks (CNN) has significantly advanced the progress in computer vision. Image classification remains foundational in all areas where CNN has demonstrated its strength. The ability to accurately classify images into predefined classes is the foundation of many real-world applications, from self-driving cars to healthcare diagnostics. For instance, in healthcare, a study on malaria diagnosis illustrated the robustness of a CNN-based image analysis model in categorizing single cells in thin blood smears as either infected or uninfected, achieving an accuracy of 97.37% [1]. Furthermore, CNNs have shown exceptional capabilities in classifying hyperspectral images, successfully addressing the issues associated with highly correlated spectral bands and the lack of ample training samples [2]. Within the realm of document analysis, deep CNN has established a new benchmark in document image classification and retrieval. They demonstrate an advanced ability to transform pixel-level inputs into compact representations through hierarchical abstractions, outperforming widely used handcrafted alternatives [3].

Additionally, the combination of CNNs with Recurrent Neural Networks (RNNs) and Canonical Correlation Analysis (CCA) has been effectively utilized for classifying images of white blood cells. This approach tackles challenges like overlapping multiple cells and attains greater accuracy than other leading-edge techniques [4]. These examples highlight the potential of CNNs to advance image classification methods, opening avenues to enhance performance in real-world applications spanning diverse domains.

The advent of deep learning has brought about a significant evolution in computer vision, paving the way for advanced image and data interpretation methods. At the forefront of this progress, CNNS have become powerful tools for many image analysis tasks, thanks to their ability to learn hierarchical features from data autonomously. The core of adequate training of deep networks is the optimization algorithm. Among these, the Adam optimizer has been widely acknowledged for its efficacy, demonstrating accelerated convergence in CNN training as discussed by various scholars [5-7]. On the other hand, Stochastic Gradient Descent (SGD) stands as a robust choice in scenarios entailing large-scale data, with discussions by various scholars underscoring its robustness in such contexts [8, 9].

Data augmentation plays a crucial role in the training of CNN. This process, which involves artificially expanding the training dataset, improves the model's ability to generalize across diverse data sets. One notable technique in this context is noise injection, which has gained recognition as an essential method for data augmentation in CNN training [9]. An extensive survey on modern data augmentation approaches elucidates the primary goal of increasing the volume, quality, and diversity of training data, which in turn facilitates improved model performance [10-12].

Despite significant progress, specific challenges remain, especially when data is scarce or highly imbalanced. For example, some scholars have explored alternative optimization paradigms or investigated more straightforward but effective model architectures to overcome these obstacles.

As shown in the examples above, the careful integration of CNN architectures, optimization algorithms, and data augmentation strategies continues to advance the field of computer vision. This burgeoning body of work continuously reveals the complex interplay of these components, moving towards more robust and efficient image classification models.

This paper centers on optimizing CNN utilizing the Fruits 360 dataset via three experiments. The primary objective of these experiments is to assess the influence of various optimization techniques on the model's performance, with a particular emphasis on evaluating metrics such as runtime and accuracy.

Initial experiments focus on integrating Spatial Transformer networks (STNs) as a preprocessing step to enhance spatial transformation invariance. Then, the second experiment delves into the use of the SGD optimization algorithm, evaluating its robustness in training deep networks. The final experiment evaluates the effectiveness of the VGG architecture in enhancing the CNN performance on the Fruits 360 dataset. These three experiments attempt to provide a comprehensive understanding of the relative merits of optimization strategies, both from a running time and accuracy perspective, thus enriching the broader narrative of optimizing CNN for image classification tasks.

# 2. Method

## 2.1. Dataset Preparation

The dataset utilized for training and evaluating the CNN models originates from the publicly available Fruits 360 dataset. This dataset comprises diverse fruit and vegetable images categorized into numerous classes. The dataset contains over 90,000 photos, with each image being a 100×100-pixel RGB image.

Regarding preprocessing, the images were normalized to adjust pixel values to a [0,1] scale. To boost the models' robustness and generalization capabilities, data augmentation methods like random shearing, zooming, horizontal flipping, and rotation were employed on the training images. The ImageDataGenerator class from the Keras library facilitated the implementation of this preprocessing and augmentation pipeline.

## 2.2. Convolutional Neural Network

The core principle of CNNs is to identify salient features in images without human intervention autonomously. This self-sufficient feature extraction is accomplished by a multi-layer architecture consisting of convolutional layers and pooling layers. These layers work together to extract a spatial hierarchy of features, from simple edge detections to more complex patterns, by applying various filters to progressively abstract the image into a form the network can use to make decisions. This process enables CNN to learn and generalize from visual data, recognizing and classifying images based on the known features.

The fundamental architecture of the CNN model includes three convolutional layers with escalating filter sizes (32, 64, and 128), each accompanied by max-pooling layers. Following these layers, there's a flattening step and a densely connected layer with 512 units. A dropout layer with a 0.5 rate is implemented to counteract overfitting, preceding the final softmax classification layer, which generates the probability distribution across the classes.

### 2.3. Optimization Algorithms

Two optimization techniques were explored to enhance the performance of the base CNN model:

SGD is a fundamental optimization algorithm used to minimize the cost function in various machine learning algorithms, especially deep learning. By incrementally updating the parameters based on the gradient of the loss function for a single training sample, SGD can reduce the error between the prediction and the actual output. Adding momentum to this process, expressly set to 0.9, enables the parameter update to have a memory of past gradients. This memory helps smooth the update by considering the previous direction of the parameter trajectory, thus suppressing fluctuations and speeding up convergence. This makes momentum SGD very effective in dealing with steep valleys of error surfaces with inconsistent gradients, helping to reach the optimal solution with a more stable and faster path.

STN introduces a new method to preprocess the data in CNN. Traditional CNNs are good at handling translation invariance but need help with more complex transformations such as scaling and rotation. STN addresses this problem by inserting a learnable module into the network that explicitly spatially transforms the feature maps. STN performs a three-step process: first, it generates a localization network to predict the transition parameters, then creates a grid from these parameters, and finally applies a sampler that uses the grid to produce the transition output [13, 14]. The output aims to enhance the attention of the CNN to relevant features within the image, making it less sensitive to the location and orientation of objects within the field of view. This leads to improved performance on tasks where recognizing objects despite variations in their pose or scale is crucial.

## 3. Results and Discussion

This study begins an exploratory journey to evaluate the impact of STN and SGD optimization techniques on a CNN trained with the Fruits 360 dataset. The benchmark for this study is set by the initial performance of the unoptimized CNN, which achieves 97.84% accuracy and 0.0999 loss after 50 epochs of training. This baseline performance provides an important reference point for understanding the improvement brought by each optimization technique. The iterative change curves of accuracy and loss are shown in Fig. 1.

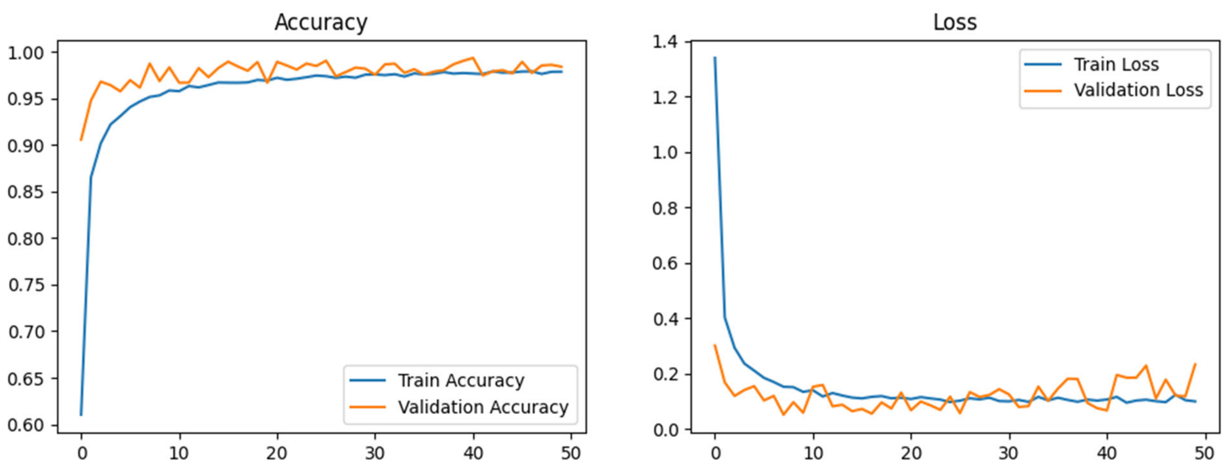


**Figure 1.** CNN's accuracy and loss curves (Photo credit: Original).

### 3.1. Influence of STN Optimization

After the introduction of STN into the CNN model, the accuracy has a marginal improvement, reaching 97.92%, and the loss slightly decreases, reaching 0.0994. However, a remarkable aspect of this result is the decrease in validation accuracy from 98.37% to 97.98% at the baseline. In the context of improved training accuracy, the slight reduction in validation accuracy indicates that STN enhances the model's generalization ability. This suggests that STN helps the CNN model adapt better to new, unseen data, although there is a slight trade-off in achieving the highest possible accuracy on the validation set. This discovery is significant as it underscores the role of STN in mitigating overfitting, a prevalent challenge in deep learning models, particularly when handling high-dimensional data like images.

The ability of STN to allow spatial manipulation of data within the network may be one of the reasons for this observed improvement. By making the network focus on the most relevant parts of the input image, STN enhances the ability of the model to learn more robust features. This is particularly beneficial for datasets like Fruits 360, which consists of various fruit images, where key identifying features may differ significantly from sample to sample. The iterative change curves of accuracy and loss are shown in Fig. 2.

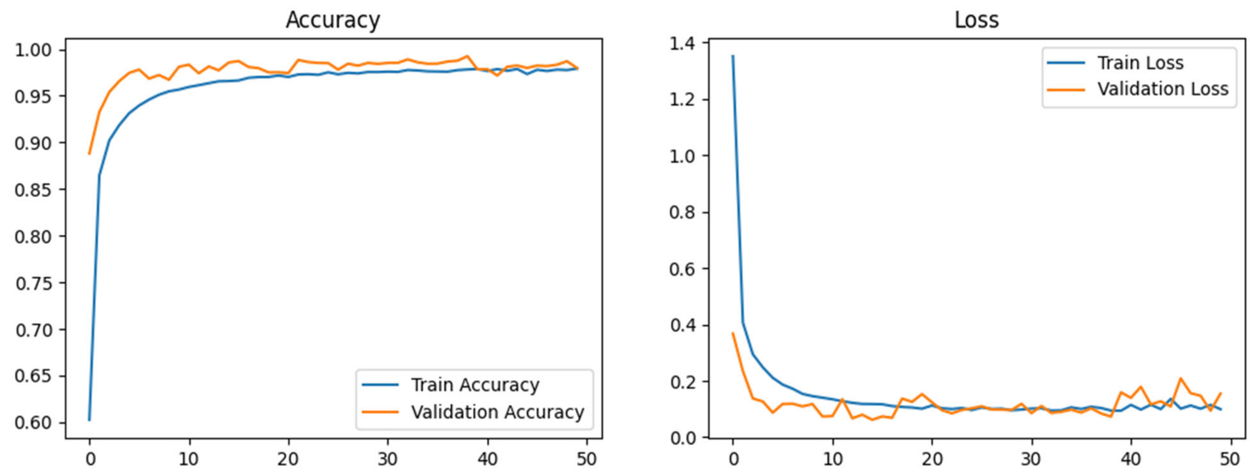


**Figure 2.** STN’s accuracy and loss curves (Photo credit: Original).

### 3.2. Efficacy of SGD Optimization

After applying the SGD optimization algorithm, the model's performance has been improved more significantly. The accuracy is improved to 98.19%, and the loss is significantly reduced to 0.0537. Moreover, there is an impressive improvement in the validation accuracy to 98.40%, and the validation loss drops to 0.0542. These results show that SGD substantially impacts on the ability of CNNS to fine-tune their parameters effectively. The significant improvement in training and validation metrics indicates that SGD can effectively improve the model's overall accuracy and its ability to generalize to new data.

SGD's efficacy is linked to its iterative method of progressing toward the optimum by evaluating the gradient of the loss function. This method is notably efficient for extensive datasets such as Fruits 360, enabling more precise adjustment of the model's weights and enhancing accuracy and generality. SGD trims down training time to 366 seconds per run, in contrast to 393 seconds for the baseline model, highlighting its efficiency as an optimization technique. This aspect is vital in real-world applications due to the importance of training time and computational resources. The iterative accuracy and loss change curves are depicted in Fig. 3.

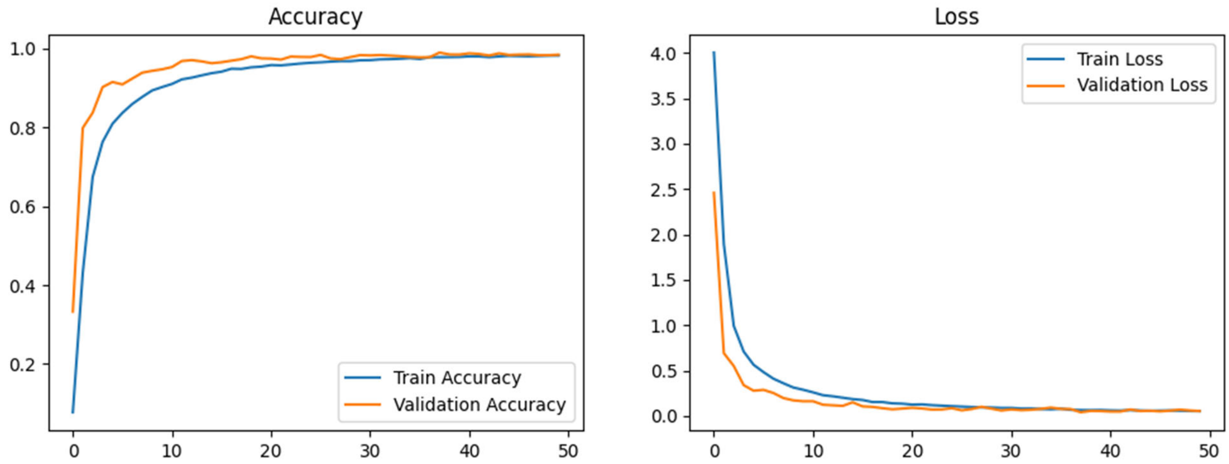


**Figure 3.** SGD's accuracy and loss curves (Photo credit: Original).

### 3.3. Comparative Analysis and Synthesis

As can be seen from Table 1, when comparing STN and SGD, it is clear that these two methods have a clear advantage in optimizing CNN. The strength of STN lies in its ability to enhance generalization and reduce overfitting, making it a valuable tool in scenarios where model robustness is critical. In contrast, SGD stands out for its ability to significantly improve model accuracy and reduce loss, making it a balanced option for comprehensive model optimization.

**Table 1.** Accuracy and Loss

| Model | CNN | STN | SGD |
|---|---|---|---|
| Accuracy | 0.9784 | 0.9792 | 0.9819 |
| Loss | 0.999 | 0.0994 | 0.0537 |

These two optimization techniques should be based on the specific requirements of the task at hand. STN would be preferred for applications that slightly compromise on maximum accuracy for better generalization. On the contrary, for functions that require the highest precision and efficiency, SGD is more suitable.

However, it is crucial to acknowledge the constraints inherent in this study. The experiments were conducted on a singular dataset, and the outcomes might exhibit variations when applied to diverse datasets or more intricate image classification tasks. Furthermore, the study did not delve into the potential synergies arising from the combination of STN and SGD, which could potentially enhance accuracy and generalization capabilities. Subsequent research endeavors should aim to overcome these limitations by extending the application of these optimization techniques to a broader spectrum of datasets and real-world scenarios. Moreover, exploring the collaborative utilization of STN and SGD has the potential to unveil fresh insights, fostering higher performance in CNN models.

## 4. Conclusion

Exploration of STN and SGD Optimization Techniques This study yielded essential insights into enhancing CNN performance. Although both techniques improve the accuracy and efficiency of CNN models trained on the Fruits 360 dataset, they exhibit different advantages and trade-offs. STN emerged as a robust method for enhancing generalization, demonstrating its ability to make the model less prone to overfitting without significantly affecting accuracy. This property makes STN particularly valuable in scenarios where the ability to generalize from limited data is critical. However, the slight decrease in validation accuracy indicates a delicate balance between generalizing and achieving the highest possible accuracy.

On the other hand, SGD shows advantages in significantly improving the accuracy and loss metrics. Its impact is more pronounced, fine-tuning the model's parameters for better performance on the training data and enhancing its generalization ability, as evidenced by the improved validation accuracy and loss. This balance of precision and adaptability makes SGD a powerful tool for tasks where both high accuracy and high efficiency are critical.

The comparative analysis of these two methods highlights an essential aspect of deep learning model optimization: no one-size-fits-all solution exists. The choice between STN and SGD should be guided by the specific demands of the application and the inherent trade-offs each method presents. This study underscores the necessity of a thoughtful selection of optimization techniques based on the dataset and the model's intended use.

Despite the promising results, this study has limitations. Conducted on a single dataset, the generalizability of the findings to other types of data or more complex tasks remains to be tested. Future research should extend these findings to a broader range of datasets and explore the potential of combining STN and SGD. Such a hybrid approach could harness the strengths of both optimization techniques, paving the way for more sophisticated and efficient CNN models.

## References

[1] Cheng Liang, Z., Powell, A., Ersoy, I., Poostchi, M., Silamut, K., Palaniappan, K., ... & Thoma, G. CNN-based image analysis for malaria diagnosis. In 2016 IEEE international conference on bioinformatics and biomedicine (BIBM) (pp. 493 - 496). IEEE, 2016.

[2] Yu, S., Jia, S., & Xu, C. Convolutional neural networks for hyperspectral image classification. Neurocomputing, 219, 88 - 98, 2017.

[3] Harley, A. W., Ufkes, A., & Derpanis, K. G. Evaluation of deep convolutional nets for document image classification and retrieval. In 2015 13th International Conference on Document Analysis and Recognition (ICDAR) (pp. 991 - 995). IEEE, 2015.

[4] Patil, A. M., Patil, M. D., & Birajdar, G. K. White blood cells image classification using deep learning with canonical correlation analysis. Irbm, 42 (5), 378 - 389, 2021.

[5] Reyad, M., Sarhan, A. M., & Arafa, M. A modified Adam algorithm for deep neural network optimization. Neural Computing and Applications, 1 - 18, 2023.

[6] Kingma, D. P., & Ba, J. Adam: A method for stochastic optimization. arXiv preprint arXiv: 1412.6980, 2014.

[7] Ogundokun, R. O., Maskeliunas, R., Misra, S., & Damaševičius, R. Improved CNN based on batch normalization and adam optimizer. In International Conference on Computational Science and Its Applications (pp. 593-604). Cham: Springer International Publishing, 2022.

[8] Dubey, S. R., Singh, S. K., & Chaudhuri, B. B. AdaNorm: Adaptive Gradient Norm Correction based Optimizer for CNNs. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 5284 - 5293), 2023.

[9] Akbiyik, M. E. Data augmentation in training CNNs: injecting noise to images. arXiv preprint arXiv: 2307.06855, 2023.

[10] Mumuni, A., & Mumuni, F. Data augmentation: A comprehensive survey of modern approaches. Array, 100258, 2022.

[11] Poojary, R., Raina, R., & Mondal, A. K. Effect of data-augmentation on fine-tuned CNN model performance. IAES International Journal of Artificial Intelligence, 10 (1), 84, 2021.

[12] Nanni, L., Paci, M., Brahnam, S., & Lumini, A. Feature transforms for image data augmentation. Neural Computing and Applications, 34 (24), 22345 - 22356, 2022.

[13] Qiu, Y., Wang, J., Jin, Z., Chen, H., Zhang, M., & Guo, L. Pose-guided matching based on deep learning for assessing quality of action on rehabilitation training. Biomedical Signal Processing and Control, 72, 103323, 2022.

[14] He, X., & Chen, Y. Optimized input for CNN-based hyperspectral image classification using spatial transformer network. IEEE Geoscience and Remote Sensing Letters, 16 (12), 1884 - 1888, 2019.