An Evolutionary Approach to Automated Class-Specific Data Augmentation for Image Classification

Silviu Tudor Marc(✉), Roman Belavkin, David Windridge, and Xiaohong Gao

Department of Computer Science, Middlesex University, London NW4 4BT, UK
SM2947@live.mdx.ac.uk

Abstract. Convolutional neural networks (CNNs) can achieve remarkable performance in many computer vision tasks (e.g. classification, detection and segmentation of images). However, the lack of labelled data can significantly hinder their generalization capabilities and limit the scope of their applications. Synthetic data augmentation (DA) is commonly used to address this issue, but uniformly applying global transformations can result in suboptimal performance when certain changes are more relevant to specific classes. The success of DA can be improved by adopting class-specific data transformations. However, this leads to an exponential increase in the number of combinations of image transformations. Finding an optimal combination is challenging due to a large number of possible transformations (e.g. some augmentation libraries offering up to sixty default transformations) and the training times of CNNs required to evaluate each combination. Here, we present an evolutionary approach using a genetic algorithm (GA) to search for an optimal combination of class-specific transformations subject to a feasible time constraint. Our study demonstrates a GA finding augmentation strategies that are significantly superior to those chosen randomly. We discuss and highlight the benefits of using class-specific data augmentation, how our evolutionary approach can automate the search for optimal DA strategies, and how it can be improved.

Keywords: Data Augmentation · Genetic Algorithm · Regularization · Hyperparameter Optimization

1 Introduction

As a field of study, computer vision strives to equip computers with the ability to accurately comprehend and efficiently process visual data, such as images and videos [6,27]. Computer vision involves several sub-domains, such as scene reconstruction, object detection and recognition, and image restoration. Convolutional neural networks (CNNs), a specialized variant of deep feedforward networks, have gained widespread popularity as a powerful technique in computer vision

https://doi.org/10.1007/978-3-031-50320-7_12

[12]. These networks have demonstrated remarkable potential for achieving high generalization accuracy in classification tasks, as evidenced by their success in various applications such as high-resolution remote sensing scene classification [15], fashion item classification [11], facial recognition [16] and numerous other use cases. Among these, CNNs are used in the field of computer-aided diagnoses (CAD), such as liver lesion classification [10] and detecting lung diseases in X-ray images [3]. A CNN learns to identify patterns and objects in the data using a labelled dataset. This process involves iteratively updating the weights in the network so that the model can learn to classify images or data samples into different categories accurately. This training process aims to develop a CNN model that can generalize well to new data samples that it has not seen before. The capability of a CNN to apply its learned knowledge from labelled datasets and make predictions on new, unseen data is called generalization.

It is well-known that the generalization ability of a CNN is in direct relationship with the size and quality of the dataset. In this case, as the size and quality of the dataset used to train a CNN increase, the network's generalisation ability also increases in a directly proportional manner [20]. However, many applications have difficulties with the data collection process, as sometimes data is scarce or expensive to label [29]. A surrogate for data collection is data augmentation (DA), which increases the dataset synthetically. DA facilitates creating an artificially expanded training set by generating modified data from existing data points [29]. Two main types of DA commonly used in computer vision are global DA and class-specific DA. The former involves applying the same transformations to all images in the dataset, regardless of their class. Several studies have researched and proposed global DA strategies that improved the robustness and generalization ability of deep feedforward networks [24]. Although this can be beneficial in some cases, it can also result in suboptimal performance due to the differential class relevance of different transformations [13].

On the other hand, class-specific DA applies data transformations designed explicitly for each class in the dataset. The transformations used for each class may vary depending on the specific characteristics of the class, such as object orientation or lighting conditions [26]. Class-specific approaches to DA may be beneficial, which has been empirically demonstrated in several cases [13,25] and [21]. DA can be considered an iterative process because it involves testing different augmentation techniques (e.g. rotation, contrast) and parameters (e.g. angles, limit) to see how they affect the performance of a machine-learning (ML) model. This process typically involves trial and error, as different datasets and models may require different augmentation methods and parameters to achieve optimal performance [23]. Various augmentation libraries, including Albumentations [4], provide an extensive selection of image transformation techniques, with over sixty options available. The availability of such a vast range of techniques further underscores the iterative nature of DA, as it may be necessary to evaluate multiple methods before arriving at an optimal approach for a given task. Additionally, implementing class-specific DA becomes more complex as the number of potential combinations increases exponentially with each additional class in the dataset. This exponential growth in possible combinations highlights the challenges of developing effective DA strategies for multi-class classification problems. The search space of class-specific DA can be computed as α^l, where α is the number of image transformations in a library, and l is the number of classes in the dataset. As a result, we are confronted with a potentially intractable combinatorial problem over a finite set of transformations. Being intrinsically combinatorial, no immediate gradient descent procedure is available, so the problem consequently cannot be absorbed into the underlying ML problem, being thus one of hyperparametric optimization. For this reason, metaheuristics were selected as our preferred approach. Metaheuristics represent a class of optimization algorithms adept at finding approximate solutions to complex optimization problems [18]. These algorithms are designed to explore large search spaces and identify high-quality solutions efficiently. Noteworthy examples of metaheuristic algorithms include simulated annealing, genetic algorithm (GA), ant colony optimization, particle swarm optimization, and tabu search. The diverse metaheuristic approaches make them popular for addressing optimization problems in various domains.

GA's are an evolutionary algorithm that takes inspiration from genetic principles, particularly natural selection. A GA utilizes various operators, including selection, mutation, and recombination, to mimic the evolutionary process and ultimately find an optimal solution to a given problem [2]. The application of evolutionary algorithms for global DA has been investigated in prior research such as [28] and [23]. Concurrently, various techniques for class-specific DA without employing any search algorithm have been suggested and have exhibited advancements in the capacity for generalization, such as those presented in [25] and [21].

To the best of our knowledge, there has been no prior research on using GA to address class-specific DA problems. Consequently, we introduce an automated search approach for class-specific DA in a classification task, utilizing a GA. The proposed framework comprises two algorithmic components: a GA and a CNN. The GA will be utilized to search for class-specific DA strategies doubling the dataset, while the CNN will be trained to extract features from the augmented dataset. The GA and CNN will work together in a fitness score-based testing approach, where the CNN will test the solutions found by the GA and provide a fitness score based on the final model generalization ability.

We present the results of our proposed framework on three datasets from the e-commerce and healthcare industries. The results demonstrate the ability of the GA to find better class-specific augmentation strategies with an improvement of up to 62.57% and up to 36.70% on the fashion and medical dataset, respectively, than the manually set global augmentation strategy. The results provide strong evidence of the effectiveness of our framework in improving the performance of CNN in image classification tasks.

The following structure is adhered to in the subsequent sections of this article. Initially, we provide an overview of various frameworks associated with class-specific DA and evolutionary techniques for automating the DA process. Subsequently, we introduce our methodologies and datasets. We then describe the experimental setups and metrics employed for evaluation. Lastly, we present the performance of our approach in the results section.

The code can be found at
https://anonymous.4open.science/r/EvoStarComparison-32E1/.

2 Related Works

Previous studies have proposed evolutionary-based global data augmentation (DA) frameworks. For example, Terauchi et al. [28] applied a thermodynamical genetic algorithm (GA) to an existing global auto-augmentation method (AutoAugment [5]), and proposed a solution to controlling the diversity of genotypes in the population. Their findings showed comparable improvement in accuracy while significantly reducing the search space and time of the search. Another example is presented in [23], where an evolutionary-based automatic DA search tool is developed to explore optimal global augmentation strategies. Despite the automation of the DA process, their findings indicate that the implemented strategies did not exhibit statistically superior performance compared to the manual strategies.

In addition, class-specific DA frameworks have been proposed. For instance, in [25], semantic attacks are used to generate new data in the case of object detection, where a small but imperceptible perturbation is introduced to the real-world data to cause the model to predict incorrectly. The model is then trained on the samples generated by the semantic attacks, resulting in improved average precision of a specific class and the overall map of the object detection network. Another study demonstrated that class-specific DA could improve ML performance on datasets with limited data, [21]. This paper's remaining sections will present our methods, datasets, experimental setups, and results.

3 Methods

In deep learning (DL), it is widely acknowledged that deep feedforward neural networks require a substantial amount of labelled data to achieve high levels of accuracy. As a result, the generalization performance of a convolutional neural network (CNN) is correlated with the size of the dataset [20]. Nevertheless, acquiring new real-life data is infeasible, arduous, or cost-prohibitive in numerous cases. For instance, the medical field may encounter such issues when dealing with rare diseases where data is scarce. Another example, companies may face obstacles when attempting to leverage DL models due to the expense or difficulty of generating or labelling datasets [9]. Data augmentation (DA) is a cost-effective approach that enables the synthetic enlargement of a dataset by manipulating the existing data. DA typically involves the following steps:

1. Selecting the augmentation methods: These may include rotation, translation, scaling, flipping, colour shifting, etc.

2. Applying the augmentation methods: This involves using libraries to automatically generate new examples by applying the selected transformations to the original images.
3. Adding the augmented data to the dataset: A more extensive dataset can be used to train the ML model, potentially leading to better performance and more accurate predictions.
4. Evaluate the performance of the ML model trained on the augmented dataset and compare it to the model trained on the original dataset.
5. Repeat 1 to 4 steps until we are satisfied with the model's performance.

Previous studies have shown that class-specific DA can improve the generalization performance of CNNs. The principal advantage of utilizing class-specific DA compared to global augmentation strategies is that the former facilitates a more individualized DA approach customized to each class in the dataset. Consequently, this may confer an edge regarding generalization capability and increased accuracy compared to a global DA approach [26]. This study focuses on automating the process of class-specific image DA in the context of multi-class classification. It utilizes a CNN as a tool for image analysis and a GA for searching for the best DA strategies.

Let us estimate the complexity of this problem. Let A be the set of all image transformations used in our study $A = \{a_1, \ldots, a_\alpha\}$. Here, α is the total number of such transformations. If the dataset contains l classes, the total number of possible class-specific DA transformations is α^l. Here we utilize Albumentations [4], a pre-existing library that provides access to more than $\alpha = 60$ diverse image transformation techniques. The datasets used in our study contain $l = 3$ and $l = 2$ distinct classes. Thus, the total size of the search space is $\alpha^l = 60^3 = 216,000$ and $60^2 = 3,600$, respectively.

Although this may not appear like a very large search space, the average time required to obtain a single solution is approximately 4.55 min, which involved training and testing a CNN on our system (Intel(R) Xeon(R) Platinum 8268 CPU 2.90 GHz, 132 GB of RAM, and a Quadro RTX 8000 GPU with 45556MiB using CUDA 11.2). It is easy to check that the total time required to test all possible solutions is approximately $216,000 \times 4.55\,\mathrm{m} \approx 1.87$years. Due to the enormous combinatorial challenge presented by the $216,000$ possible class-specific DA strategies and the associated time required to solve the problem ($\approx$ 2 years), a naive approach is deemed impractical. In computational optimization, evolutionary algorithms have been widely acknowledged as efficient for large combinatorial problems, thus offering a feasible solution to this challenge [8].

To address this, we present a framework that integrates two crucial elements: a GA and a CNN, that function in a mutually beneficial fitness-score relationship to determine the optimal class-specific DA strategy. The GA plays a pivotal role in automating the selection of class-specific DA strategies required to augment the original dataset. The resulting augmented dataset is then utilized for training the CNN architecture, with the validation loss serving as the fitness score, guiding the GA's evolution process.

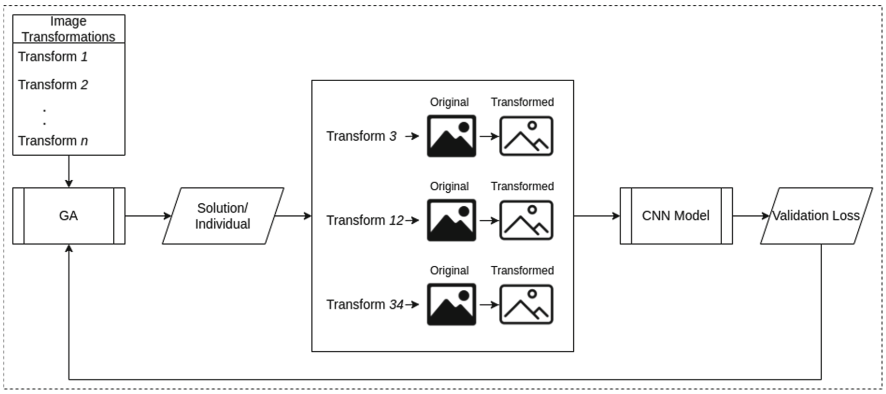

Fig. 1. The proposed framework starts from the GA generating a random population from a given list of image transformations. An individual contains l augmentations where l is the number of classes in the dataset (e.g. 3). The augmentations are used to double the size of the dataset. Then the dataset is fed into the CNN model, outputting a validation loss which we use to calculate and assign a fitness score to the individual.

The workflow of the proposed framework, as seen in Fig. 1, is composed of the following steps:

1. The CNN is trained on the original datasets. This process is repeated ten times, and the resultant baseline score is obtained by averaging the scores from these training sessions.
2. Next, the GA is employed to automatically generate a set of class-specific DA strategies, which we call solutions.
3. For each class-specific DA strategy (solution) produced by the GA, a new dataset is constructed by combining the baseline dataset with the augmented dataset, thus doubling the number of images.
4. The CNN is then trained on the augmented dataset, and metrics such as accuracy and loss are recorded.
5. The subsequent evaluation of the results involves comparing them with the average baseline score to determine the effectiveness of the proposed framework.

This framework offers a powerful solution for automating the selection of DA strategies and improving the performance of image classification models. The results of this study will contribute to a deeper understanding of the capabilities of this framework and inform future research in this area.

In the subsequent portion of this section, we will expound upon the CNN architectures used, datasets, pre-processing steps, and the representation of the GA and its operators.

3.1 CNN

In the present investigation, we employed two distinct CNN architectures: ResNet18 [14], an off-the-shelf architecture, and a customized architecture modelled after an architecture described in [23].

In the case of ResNet18, we also utilized the pre-trained weights made available by the PyTorch library [22]. To adjust the ResNet18 architecture to our datasets, we replaced the last fully connected layer with a novel layer with the same number of input features as the original model and the number of classes in our dataset. Our study employed the categorical cross-entropy loss function as our optimization objective, quantifying the disparity between the predicted class probabilities and the actual class labels [19]. The PyTorch library afforded us access to a pre-existing sparse categorical cross-entropy criterion [22]. The CNN architectures were optimized using the Adam and AdaDelta optimizers [17], which are well-suited for fine-tuning deep learning models and use gradient descent to update the model parameters, with additional features such as adaptive learning rates and momentum to improve convergence.

3.2 Datasets and Pre-processing

To guarantee the versatility of the proposed framework across diverse applications, we conducted experiments on three datasets originating from various domains, such as medicine and fashion e-commerce. The initial dataset employed in this study is MESO [7], which comprises tissue microarrays stained with hematoxylin and eosin, two commonly utilized staining agents to enhance the visualization of cells in biopsies. The dataset is composed of 243 cores, of which 155 are epithelioid, 64 are biphasic, and 24 are sarcomatoid in type. The second dataset employed in this study is the Fashion dataset, a subset of the DeepFashion2 dataset [11]. This dataset aims to distinguish between three comparable classes, specifically pants, joggers, and jeans. Additionally, to replicate a scarcity of data scenario, we randomly chose 300 images for each class, resulting in a total of 900 images for the complete dataset. The Breast Histopathology dataset, containing 277,524 images measuring 50×50 pixels, is the third dataset employed in this study, consisting of 198,738 IDC-negative images and 78,786 IDC-positive images. Invasive Ductal Carcinoma (IDC) is the most frequently occurring breast cancer subtype (Fig. 2).

Table 1. Number of images, classes and splits used in our experiments

Dataset	# Classes	# of samples	Split
MESO	3	243	80/10/10
Fashion	3	900	80/10/10
BreastHisto	2	58,742	70/15/15

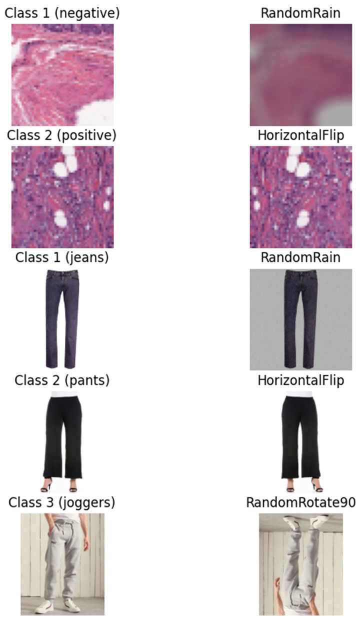

Fig. 2. Examples of original images (left) and augmented images (right).

The datasets were divided into the train, validation, and test subsets using random split, as seen in Table 1. The training subset was employed to train the model, updating the model parameters to minimize the loss function through the backpropagation of the optimisation algorithm (Adam or AdaDelta). Conversely, the validation subset was not utilized to update the model parameters but to evaluate the model's performance and prevent overfitting. The test subset was kept separate from the training and validation sets and only used once to evaluate the final model performance after the training process. To ensure compatibility with the ResNet18, all images were resized to a resolution of 224×224 pixels. The entire dataset underwent normalization with means $(0.485, 0.456, 0.406)$ and standard deviations $(0.229, 0.224, 0.225)$.

3.3 Genetic Algorithm

In this investigation, we employ a simple two-step genetic algorithm (GA) that utilizes a singular crossover operator and incorporates the application of elitism. The GA commences by producing a series of random solutions, subsequently utilized to expand the primary datasets and train the convolutional neural network (CNN). The minimum validation loss attained through CNN training is utilized to compute a fitness score subsequently assigned to each solution; see Fig. 3. The second step selects the best solutions for the crossover operation.

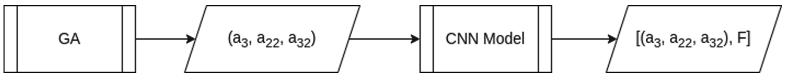

Fig. 3. Example of a GA-created solution and later scored by our CNN as found in our code, here F represents the fitness score.

During the crossover, each top solution is decomposed into its constituent transformations, which are then aggregated into a single list. New solutions are then formed randomly using this aggregated list; see Algorithm 2.

Algorithm 1 Genetic Algorithm for optimal class-specific DA

```
Require: A list of possible augmentations, target score $T$
Ensure: Optimal class-specific DA strategy
 1: Initialize population
 2: while not last generation do
 3:     Evaluate each solution and assign a fitness score $F$ ;
 4:     if $F \geq T$ then
 5:         $break$;
 6:     end if
 7:     Rank solutions from best to worst;
 8:     Select top $n$ solutions as parents;
 9:     Keep the best solution intact (elitism);
10:     Create new offspring through crossover;
11:     Replace the old population with newly created offspring and the elite individual;
12: end while
13: return Best individual
```

The GA's evolution process, as seen in Algorithm 1, is guided by a fitness function that evaluates the performance of each solution and returns a fitness score F. This is calculated as the value of the inverse of the best validation loss achieved during the training of the CNN architecture, $F = |\text{ValidationLoss}|^{-1}$. This approach allows the GA to prioritize the solutions that lead to the lowest validation loss and, consequently, to the highest fitness score.

Algorithm 2 GA's crossover operator

```
Require: List of top $n$ individuals;
Ensure: New offspring formed through crossover;
 1: Combine all image augmentations from top individuals into a single list
 2: Shuffle the list to introduce randomness;
 3: Select a random number $m$ of augmentations and form new offspring ;
 4: return New offspring list;
```

4 Experimental Setup

In this research article, we conduct two experimental setups. The first setup involves replicating the experimental conditions described in [23], where we utilize the CNN and workflow presented by the authors along with k-fold cross-validation, elitism, dataset, and using Imgaug library and image transformations. K-fold cross-validation is a widely used technique in ML and statistical modelling to evaluate a predictive model's performance and generalization ability [1]. Specifically, k-fold cross-validation entails randomly partitioning the dataset into k equal-sized subsets or folds, followed by the iterative training and testing of the model k times. As a second experimental setup, we utilize only the genetic algorithm and the crossover operator without incorporating the elitism strategy and k-fold cross-validation. Here we use the Albumentations library.

In both setups, we train the CNN architectures on the datasets from Table 1 and record their baseline performances, saving the model with an R prefix followed by a unique code (e.g., R_j2763), so we can later differentiate it from the others, which are only saved as a unique code (e.g., 2nj942). We record metrics, such as training loss and accuracy, validation loss and accuracy, confusion matrices, precision, recall, the F_1-score, best epoch, total training time, fitness score and the augmentation strategy used. However, for the evaluation of the model, we only use the validation loss and fitness score as described in Sect. 4.1.

4.1 Evaluation Metric

Validation loss is a commonly used metric in the training of neural networks. It measures how well the model can generalize to new, unseen data. The validation loss is calculated using a separate data set held out during training, known as the validation set. This set evaluates the model's performance on data not seen during training. The validation loss is calculated after each epoch using the PyTorch built-in categorical cross-entropy [22]. The validation loss is calculated by feeding the validation set through the trained model and calculating the average error between the model's predictions and the actual class labels. It can be used to track the model's performance throughout training and to identify overfitting, which occurs when the model starts to memorize the training data and performs poorly on new data. In summary, the validation loss is an important metric for evaluating the performance of a neural network, especially for avoiding overfitting and ensuring that the model generalizes well to new, unseen data (Table 2).

Table 2. Performance of GA evolution. The numbers show an average cross-entropy loss, their differences and per cent improvement relative to a baseline.

TestID	Baseline	Avg10BestSol	Diff (%)	Best Base	Best Sol	Diff (%)
T_6b7326	.3780	.3676	.0104 (2.7%)	.3750	.3669	.0081 (2.2%)
T_347958	.4027	.3685	.0342 (8.5%)	.3951	.3586	.0365 (9.3%)
T_f24029	.6873	.6844	.0029 (0.4%)	.6864	.6843	.0020 (0.3%)
T_38b1e9	.6715	.6573	.0142 (2.1%)	.6696	.6549	.0147 (2.2%)

5 Results

In this section, we present the results of our experiments. Across all three datasets examined, our research consistently reveals notable improvements over the baseline. Notably, on the BreastHisto dataset, which already featured a substantial volume of images, we observed a commendable 8.49% improvement. In stark contrast, the MESO dataset, characterized by its comparably smaller and skewed nature, exhibited a significant performance boost, registering an impressive 18.99% improvement. The comparison between the baseline validation losses and the performance of the GA is presented in Table 3. The results indicate that the GA successfully discovered many class-specific DA solutions outperforming the baseline. The table also displays the average validation losses of these improved solutions and highlights the best validation loss achieved for each experiment.

In Experiment 1, the results do not consistently exhibit significant improvements. Nevertheless, the improvements are consistent even in cases where the CNN model fails to converge. Four tests were performed, and all showed improvements ranging from 2% to 5%. The failure of the CNN model to converge is not necessarily correlated with the framework but rather with the CNN architecture used. This observation is evident in T_f24029 and T_347958, where one of the main reasons for the difference between the average baselines (.6873 and .4027) could be the CNN architecture or the optimizer. Conversely, the average baselines are similar when the same CNN architecture is used, such as in T_347958 and T_6b7326 (.4027 and .3780). The results of T_6b7326 suggest that class-specific DA outperforms global DA, as the baseline training was conducted on a randomly oversampled dataset. The highest improvement ratio is observed in T_347958, with an improvement of 0.0342 over the baseline score.

The graphs presented in Fig. 4 and Fig. 5 illustrate the average fitness per generation, demonstrating the steady improvement of our Genetic Algorithm (GA) in enhancing the overall population fitness score over time. In test T_347958 Fig. 5, we observe the progress of a 5-generation GA, while in test T_6b7326, a 20-generation GA is displayed. In both cases, the populations evolve towards enhanced performance, but it's evident that increasing the number of generations results in more substantial performance improvements (Fig. 6).

In contrast, the results of Experiment 2 demonstrate more significant improvements, ranging from 0.1537 to 0.2616, representing a substantial improvement. We conducted four distinct experiments, all of which produced satisfactory outcomes. However, in T_e61c77, the dataset is exceptionally small, necessitating oversampling three or four times. Nevertheless, our framework could still identify better class-specific DA strategies than global DA. If the oversample baseline hyperparameter is true, it implies that the baseline validation was conducted on a randomly augmented dataset at the global level. We do this to simulate the manual selection of global DA and train the CNN on the same number of images.

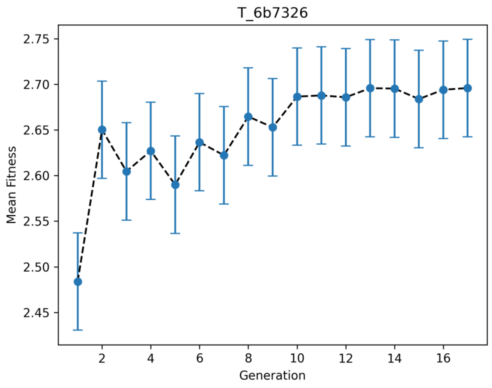

Fig. 4. Mean and STD of fitness over generations in Experiment 1 test T_6b7326.

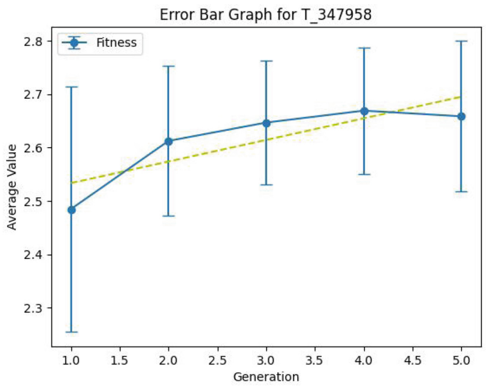

Fig. 5. Mean and STD of fitness over generations in Experiment 1 test T_347958.

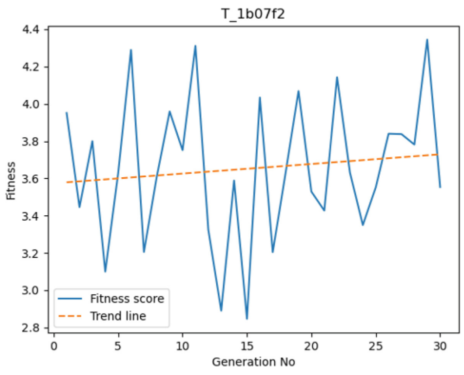

Fig. 6. Mean of fitness over generations in Experiment 2 test T_1b07f2.

Table 3. Experiment hyperparameters setup results, comparison

Hyper-parameter	T_6b7326	T_347958	T_f24029	T_7749c9	T_e61c77	T_fc8b83	T_1b07f2
Dataset	Breast	Breast	Breast	Fashion	MESO	Fashion	Fashion
CNN	ResNet18	ResNet18	EvoCNN	ResNet18	ResNet18	ResNet18	ResNet18
Optimiser	Adam	Adam	AdaDelta	Adam	Adam	Adam	Adam
Resolution	24	24	32	224	224	224	224
Population Coef	2	2	1	2	2	1	1
Generations	20	10	20	30	30	20	30
Top solutions	10	10	5	10	10	10	10
Offsprings	15	10	10	10	10	10	10
Oversample	True	False	False	True	True	False	False
Pre-trained	False	False	False	True	True	True	True
K-Fold	True	True	True	False	False	False	False
Experiment No	1	1	1	2	2	2	2
Data split	70/15/15	70/15/15	70/15/15	80/10/10	80/10/10	80/10/10	80/10/10
BaselineAvg	.3780	.4027	.6873	na	na	na	na
Avg10BestSol	.3676	.3685	.6844	.1718	.5773	.1677	.1481
Diff	.0104	.0342	.0029	na	na	na	na
(%)	(2.75%)	(8.49%)	(0.42%)	na	na	na	na
Best Baseline	.3750	.3951	.6864	.3639	.7127	.2968	.2944
Best Solution	.3669	.3586	.6843	.1426	.4511	.1431	.1102
Diff	.0081	.0365	.0020	.2213	.2616	.1537	.1842
(%)	(2.16%)	(9.23%)	(0.29%)	(60.81%)	(36.70%)	(51.79%)	(62.57%)
Time(h)	90.45	56.04	35.91	31.63	26.89	23.42	23.75

Finally, another substantial achievement is the time spent to find this improvement which does not exceed 3.75 days. Comparing this with the total time that would be needed to use the brute force approach 682.5 days results in 182% improvement in search efficiency.

6 Discussion

The study's conclusion highlights the success of the proposed automatic framework in finding class-specific data augmentation strategies using a combination of a genetic algorithm and a convolutional neural network. The improvement in validation loss compared to the baseline results indicates that the framework effectively optimizes data DA strategies for improved CNN performance. The proposed framework can potentially impact various applications in the e-commerce and healthcare industries, where accurate and robust image classification models are crucial. Automating the search for class-specific data augmentation strategies can save time and effort compared to manual experimentation with different augmentation strategies.

However, some limitations to the proposed framework should be considered. For instance, the framework may not be suitable for all data types and may require modifications for more complex and diverse data types. Furthermore, when working with medical images, it is imperative for medical specialists to validate the strategies discovered by the GA. In light of these limitations, future work should focus on extending the framework to handle more complex and diverse data types. This can help to improve the framework's performance and applicability to a broader range of image classification problems.

Overall, the proposed framework represents a promising approach to automating the search for class-specific data augmentation strategies, and the results of this study demonstrate its effectiveness in improving CNN performance.

7 Conclusion

In conclusion, this study introduces an automated framework to identify class-specific data augmentation strategies through a genetic algorithm and a convolutional neural network. The results demonstrate that the proposed framework successfully found superior class-specific data augmentation strategies compared to the manually set global augmentation strategy. The two experiments showed that the genetic algorithm was able to achieve improvements of 62.57% and 36.70% in validation loss in contrast to the baseline results. Moreover, a substantial accomplishment within this framework is the minimal time required to achieve these improvements, not exceeding 3.75 days. When juxtaposed with the total time that would have been necessitated by the brute force approach, amounting to a staggering 682.5 days, the efficiency of our framework shines. This comparison reveals an extraordinary 182% improvement in search efficiency.

These findings highlight the effectiveness of the proposed framework in enhancing CNN performance by optimizing data augmentation strategies. The proposed framework presents a promising approach for automating the search for class-specific data augmentation strategies that can benefit various industries, such as e-commerce and healthcare. Further research could extend the framework's capabilities to handle more complex and diverse data types while exploring different operators, such as controlled mutation rates for the genetic algorithm component.

References

1. Arlot, S., Celisse, A.: A survey of cross-validation procedures for model selection (2010)
2. Bäck, T., Schwefel, H.P.: An overview of evolutionary algorithms for parameter optimization. Evol. Comput. **1**(1), 1–23 (1993)
3. Bharati, S., Podder, P., Mondal, M.R.H.: Hybrid deep learning for detecting lung diseases from x-ray images. Inform. Med. Unlocked **20**, 100391 (2020)
4. Buslaev, A., Iglovikov, V.I., Khvedchenya, E., Parinov, A., Druzhinin, M., Kalinin, A.A.: Albumentations: fast and flexible image augmentations. Information **11**(2) (2020). https://doi.org/10.3390/info11020125, https://www.mdpi.com/2078-2489/11/2/125

5. Cubuk, E.D., Zoph, B., Mané, D., Vasudevan, V., Le, Q.V.: Autoaugment: learning augmentation policies from data. CoRR abs/1805.09501 (2018). http://arxiv.org/abs/1805.09501
6. Dana, H., Ballard, C.M.B.: Computer Vision. Prentice-Hall, Hoboken (1982)
7. Eastwood, M., et al.: Malignant mesothelioma subtyping of tissue images via sampling driven multiple instance prediction. In: Michalowski, M., Abidi, S.S.R., Abidi, S. (eds.) Artificial Intelligence in Medicine. AIME 2022. LNCS, vol. 13263, pp. 263–272. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-09342-5_25
8. Eiben, A.E., Smith, J.E.: Introduction to Evolutionary Computing. Springer, Berlin, Heidelberg (2015). https://doi.org/10.1007/978-3-662-43631-8_2
9. Erhan, D., Szegedy, C., Toshev, A., Anguelov, D.: Scalable object detection using deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2147–2154 (2014)
10. Frid-Adar, M., Diamant, I., Klang, E., Amitai, M., Goldberger, J., Greenspan, H.: Gan-based synthetic medical image augmentation for increased CNN performance in liver lesion classification. Neurocomputing **321**, 321–331 (2018). https://doi.org/10.1016/j.neucom.2018.09.013, https://www.sciencedirect.com/science/article/pii/S0925231218310749
11. Ge, Y., Zhang, R., Wu, L., Wang, X., Tang, X., Luo, P.: A versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images. In: CVPR (2019)
12. Goodfellow, I.J., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge, MA, USA (2016). http://www.deeplearningbook.org
13. Hauberg, S., Freifeld, O., Larsen, A.B.L., III, J.W.F., Hansen, L.K.: Dreaming more data: class-dependent distributions over diffeomorphisms for learned data augmentation. CoRR abs/1510.02795 (2015). http://arxiv.org/abs/1510.02795
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. CoRR abs/1512.03385 (2015). http://arxiv.org/abs/1512.03385
15. Hu, F., Xia, G.S., Hu, J., Zhang, L.: Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery. Remote Sens. **7**(11), 14680–14707 (2015)
16. Khan, S., Javed, M.H., Ahmed, E., Shah, S.A.A., Ali, S.U.: Facial recognition using convolutional neural networks and implementation on smart glasses. In: 2019 International Conference on Information Science and Communication Technology (ICISCT), pp. 1–6 (2019). https://doi.org/10.1109/CISCT.2019.8777442
17. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization (2014). https://doi.org/10.48550/ARXIV.1412.6980, https://arxiv.org/abs/1412.6980
18. Kirkpatrick, S., Gelatt, C.D., Vecchi, J.M.P.: Optimization by simulated annealing. Science **220**(4598), 671–680 (1983)
19. Koidl, K.: Loss functions in classification tasks. School of Computer Science and Statistic Trinity College, Dublin (2013)
20. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Commun. ACM **60**(6), 84–90 (2017)
21. Nivin, T.W., Scott, G.J., Hurt, J.A., Chastain, R.L., Davis, C.H.: Exploring the effects of class-specific augmentation and class coalescence on deep neural network performance using a novel road feature dataset. In: 2018 IEEE Applied Imagery Pattern Recognition Workshop (AIPR), pp. 1–7 (2018). https://doi.org/10.1109/AIPR.2018.8707406
22. Paszke, A., et al.: Pytorch: an imperative style, high-performance deep learning library. CoRR abs/1912.01703 (2019). http://arxiv.org/abs/1912.01703

23. Pereira, S., Correia, J., Machado, P.: Evolving data augmentation strategies. In: Jiménez Laredo, J.L., Hidalgo, J.I., Babaagba, K.O. (eds.) EvoApplications 2022. LNCS, vol. 13224, pp. 337–351. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-02462-7_22
24. Rebuffi, S., Gowal, S., Calian, D.A., Stimberg, F., Wiles, O., Mann, T.A.: Data augmentation can improve robustness. CoRR abs/2111.05328 (2021). https://arxiv.org/abs/2111.05328
25. S, A.K., Pal, A., Mopuri, K.R., Krishna Gorthi, R.: Adv-cut paste: semantic adversarial class specific data augmentation technique for object detection. In: 2022 26th International Conference on Pattern Recognition (ICPR), pp. 3632–3638 (2022). https://doi.org/10.1109/ICPR56361.2022.9956409
26. Shorten, C., Khoshgoftaar, T.: A survey on image data augmentation for deep learning. J. Big Data **6**(1), 1–48 (2019)
27. Sonka, M., Hlavac, V., Boyle, R.: Image processing, analysis, and machine vision. Cengage Learning (2014)
28. Terauchi, A., Mori, N.: Evolutionary approach for autoaugment using the thermodynamical genetic algorithm. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 9851–9858 (2021)
29. Ying, X.: An overview of overfitting and its solutions. J. Phys. Conf. Ser. **1168**(2), 022022 (2019). https://doi.org/10.1088/1742-6596/1168/2/022022, https://dx.doi.org/10.1088/1742-6596/1168/2/022022