ISSN 2319-5991 www.ijerst.com
© 2026 IJERST. All Rights Reserved
Peer Reviewed, Referred & Indexed Journal

Research Paper

Hybrid Optimization Based Hyperparameter Tuning for Enhanced Image Classification Using CNN

[1] **Gendi Penchala Narasimha,** [2] Dr B.V.S Uma Prathyusha
[1]M.Tech Scholar, Dept of ECE, Audisankara College of Engineering and Technology, Gudur, Andhra Pradesh mail id : penchalanarasimha111@gmail.com
[2]Professor, Dept of ECE, Audisankara College of Engineering and Technology, Gudur, Andhra Pradesh. mail id: bvsumaprathyusha.cse@audisankara.ac.in

Abstract**—**The hyperparameter optimization of deep convolutional neural networks is one of the most computationally intensive problems in applied machine learning, especially in the context of multi-class image classification of complex multi-class image datasets. The paper presents a hybrid optimization framework, which combines Genetic Algorithm (GA) and Jellyfish Search Optimization (JSO) with a residual Convolutional Neural Network (CNN) to automatically and optimally optimize hyperparameters to the CIFAR-10 benchmark. There are four experimental environments that are tested: a hand-crafted Custom CNN baseline, a fine-tuned MobileNetV2 transfer learning model, a GA-optimized residual CNN, and a JSO-optimized residual CNN. The JSO-CNN achieves the highest test accuracy of 91.26% and a macro F1-score of 91.23%, surpassing the Custom CNN (90.48%), the GA-CNN (89.58%), and MobileNetV2 (82.98%). These results suggest that bio-inspired metaheuristic algorithms, particularly, the Jellyfish Search Optimizer and its ocean current drift, Levy flight passive walk, and active swarm behaviour operators can be used to a great effect in comparison to manual and genetic search strategies. The proposed pipeline is complemented by a web application based on FastAPI, predicting images in real-time, which is a step towards production-ready deployment of optimized image classifiers.

Keywords: *CIFAR-10, Convolutional Neural Network, Genetic Algorithm, Jellyfish Search Optimization, hyperparameter tuning, image classification, transfer learning, MobileNetV2, metaheuristic optimization, deep learning.*

I. INTRODUCTION

Recent spurt of digital imagery in autonomous systems, medical diagnostics, satellite analysis and consumer applications has made image classification the focus of recent computer vision research. Convolutional Neural Networks have always been the paradigm of choice in terms of hierarchical spatial feature extraction of pixel arrays among the various architectural choices open to practitioners because of their parameter sharing, translational invariance, and compositional learning properties. Nevertheless, raw architectural skill is not a predictor of deployment-level performance; the selection of training hyperparameters, such as learning rate, filter dimensions, regularization coefficients, and batch size, can influence the ultimate model accuracy, which is as similar or even better than the architecture itself.

Manual hyperparameter search, also known as grid search or random search, has the disadvantage of exponential growth in the size of the configuration space, and does not offer any way to take advantage of structure in the loss landscape. Neural Architecture Search and Bayesian optimization methods partially overcome this drawback but require large computational budgets, which cannot be afforded in resource-limited academic or industrial environments. Population-based algorithms based on biologically-inspired algorithms promise a trade-off: they do not require gradient information, and scale to high-dimensional discrete search spaces, and they employ biologically-inspired diversity mechanisms that do not prematurely converge to suboptimal configurations.

The hypothesis of the paper is that a hybrid optimization framework with a strategic use of two complementary metaheuristic algorithms i.e. Genetic Algorithm and Jellyfish Search Optimization can be compared to the hyperparameter optimization of a trained residual CNN on CIFAR-10. The Genetic Algorithm is founded on evolutionary ideas of selection pressure, genetic crossover and adaptive mutation to explore the discrete hyperparameter lattice in subsequent generations.

II. RELATED WORK

A number of studies have investigated the use of hybrid optimization methods to automatically tune hyperparameters in convolutional neural networks to image classification problems, solving issues like inefficiencies in manual tuning and local optima. Another interesting approach is a PSO-GA model where Particle Swarm Optimization is used to do global exploration with velocity and position updates based on personal and global bests, and Genetic Algorithms is used to provide adaptive crossover and mutation to do refined exploitation. It typically begins with pre-processing of the data like data augmentation and normalization, and then encodes hyperparameters like learning rate, batch size, number of layers, and kernel sizes into particles or chromosomes.[1], [2].

This hybrid is used on top of pretrained models like Exception in classifying chest X-rays with COVID-19, normal, bacterial, and viral pneumonia cases in multi-class configurations with accuracies of 97.47, 99.66, and 100% respectively, and outperforms baselines by improving robustness on small medical datasets[3], [4]. Similarly, a Hybrid Particle Swarm Grey Wolf Optimizer with modified inertia-driven updates of PSO and the hierarchical leadership (alpha, beta, delta wolves) of the velocity equations of the Grey Wolf Optimization to tune convolutional layers, filter sizes, and batch sizes on MNIST, CIFAR-10, and Indian Classical Dance image data sets is statistically validated.

by Wilcoxon tests that demonstrate the superiority of the hybrid CNNs over non-hybrid CNNs in terms of sensitivity, specificity and convergence speed[5], [6].

At the same time, swarm intelligence hybrids like Particle Swarm Optimization with Artificial Bee Colony (Swarm CNN) apply a nested optimization: PSO first optimizes CNN depth and layer order with modified velocity updates that selectively keep or drop layers based on global and personal bests, and then ABC optimizes layer-specific parameters like filter counts and strides with employed, onlooker, and scout bee phases[7], [8]. This methodology trains various benchmarks such as MNIST variants, Fashion MNIST, CONVEX shapes, RECTANGLES, and CIFAR-10 with pre-processing to normalize and augment them, achieving the highest accuracies of 99.58% on standard MNIST, 100% on RECTANGLES, and 84.77% on CIFAR-10, with[9], [10].

Another variant is founded on Egyptian Vulture Optimization, which uses pebble-tossing and twig-rolling behaviours to search and angle-change mechanisms to local search to optimize brain MRI classification learning rates, momentum, batch sizes, kernel sizes, dropout rates, and activation functions. The workflow initializes solution strings, measures fitness through CNN training accuracy and repeats through biological operators until convergence on Glioma and general brain tumour datasets, providing 98.72% testing accuracy on Glioma images, and 93.51% on larger MRI sets, and 95% accuracy and lower misclassification rates than untuned MLP, DNN, or baseline CNN models[11], [12].

More recent approaches include evolutionary and multi-objective approaches, like Non-dominated Sorted Genetic Algorithm-II encoding hyperparameters as binary chromosomes to learn rates, dropout, batch sizes, number of layers, and weight init. It populates a population, uses tournament selection with crowding distance, simulated binary crossover, and polynomial mutation between generations and evaluates fitness on MNIST, CIFAR-10, CIFAR-100 classification accuracy, converging to 99.499.6% accuracies with lower training times than grid or random search.[13], [14]. A velocity-free Particle Swarm Optimization variant (PSWV) also optimizes variable-length CNN architectures by linearly combining personal and global best positions without velocity, achieving competitive accuracies on MNIST rotated/background variants and RECTANGLES datasets in a few steps, compared to 27 state-of-the-art methods. Aquila Optimizer-Harris Hawks Optimization hybrids are used in medical tasks, where the vast exploration of Aquila and local exploitation of Harris Hawks is used to optimize learning rates, filter counts and dropout on 7,023 brain MRI images in glioma, meningioma, pituitary, and non-tumour classes, with high precision, recall, F1-scores, and training[15].

III. DATASET

A. CIFAR-10 Overview

CIFAR-10 is a well-known benchmark dataset to test image classification algorithms, which was introduced by Krizhevsky and Hinton [10]. It has 60,000 colour images of 32x32 pixel resolution that are randomly spread over ten mutually exclusive semantic categories: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. The standard partition splits the images into 50,000 training and 10,000 test assessment. Each class contributes 6,000 images and this provides a perfectly balanced distribution of classes that eliminates the confounding variable of class-imbalance in comparative experiments.

Despite the simplicity of the sound, CIFAR-10 has serious recognition issues. The 32x32 pixel resolution is a significant constraint on the spatial resolution that the classifier can have. The intra-class variability of some of the categories is high due to the variability of pose, lighting and background, and the inter-class confusion is particularly high between visually similar pairs such as cat-dog and automobile-truck, which is also confirmed by the confusion patterns observed in all four experimental models in this study.

B. Preprocessing

Training images in the Custom CNN and GA/JSO-optimized models were standardized by dividing pixel values by 255.0 to scale intensities to the [0, 1] range, followed by per-channel mean and standard deviation normalization of the training split to get zero-centered unit-variance features. The minimum spatial input requirement of the pretrained backbone in the case of the

MobileNetV2 model was 32x32, which was bicubically upsampled to 96x96. Training was done with data augmentation: random horizontal flipping, small rotation up to 15 degrees, 10% width and height shifts, zoom range 10% and shear transforms. The training partition was stratified into 90/10 to be applied in validation in both the baseline training and optimization fitness evaluation.

IV. SYSTEM ARCHITECTURE

The proposed system integrates four important functional subsystems into a rational end-to-end pipeline as illustrated in Fig. 1. The computational core of the framework is the Optimization and Training Pipeline. It takes batches of raw CIFAR-10 images as input in the Data Storage layer, and optimizes hyperparameters with either Genetic Algorithm or Jellyfish Search Optimization, with candidate configurations evaluated by training lightweight proxy networks and the validation accuracy used as the fitness signal.

The Data Storage layer has three separate repositories: the CIFAR-10 image dataset available to the training pipeline, the repository of trained model artifacts, and a user credential store to support the authentication subsystem. The Web Application layer is implemented using the FastAPI framework and has two functional endpoints: a Login/Register endpoint and an Upload Image endpoint that routes the received image payloads to the Prediction API. The Prediction API loads the best suited serialized model and gives the predicted class label and confidence scores.

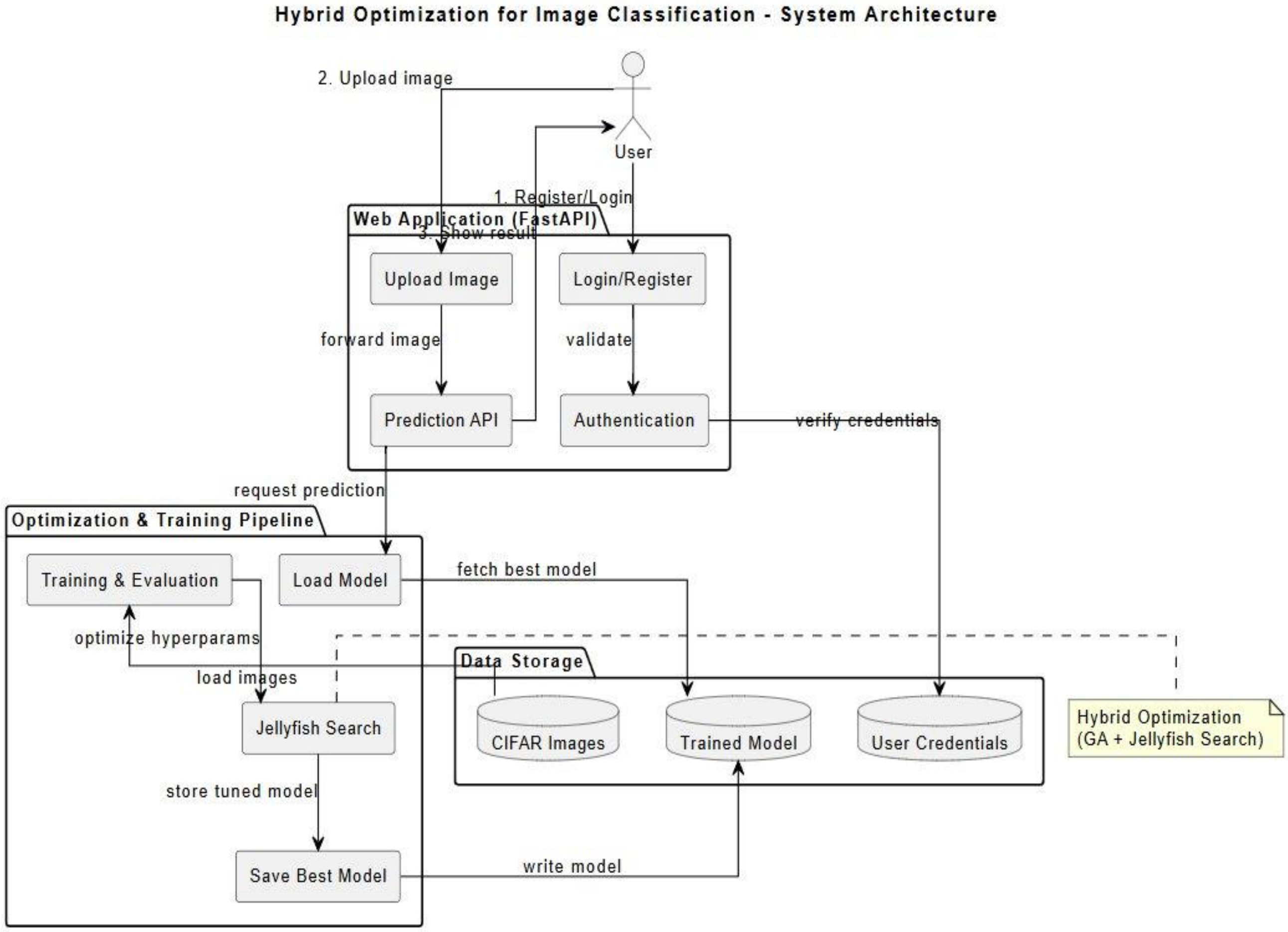


Fig. 1. Hybrid Optimization for Image Classification - System Architecture.

V. PROPOSED METHODOLOGY

A. Custom CNN Baseline

The manually engineered performance baseline is the Custom CNN. The architecture consists of three successive convolutional blocks, each having two Conv2D layers with 3x3 kernels and same-padding followed by Batch Normalization, ReLU activation, MaxPooling with 2x2 stride and Dropout regularization. The three blocks use filter counts of 64, 128 and 256 respectively with a depth of feature map that is doubled and a spatial resolution that is half of the last one. All convolutional kernels are given an L2 weight decay coefficient of 1e-4. The convolutional body is succeeded by a Global Average Pooling layer that down-samples the feature maps to a single vector, which is then fed through a Dense layer of 512 units, and the 10-class softmax classifier. The model is optimized with Adam and an initial learning rate of 1e-3 with Early Stopping patience of 10 on validation accuracy and Reduce LROnPlateau reduction factor of 0.5.

B. MobileNetV2 Transfer Learning

The MobileNetV2 model is trained in two phases with ImageNet-pretrained weights. Phase 1: The whole backbone is frozen and the classification head is trained with 15 epochs with a learning rate of 1e-4. Phase 2: The final convolutional layers of the backbone are unfrozen and the whole model is fine-tuned with a smaller learning rate using cosine annealing to make sure that the previously trained feature representations are not disastrously forgotten. The change of phases causes a temporary drop in training accuracy as the model reconfigures its backbone weights to the CIFAR-10 distribution.

C. Genetic Algorithm Hyperparameter Optimization

The Genetic Algorithm is used on a discrete hyperparameter space that is a set of filter configurations, dropout rates, learning rate values, dense unit counts, batch sizes, and optimizer choices. Each member of the population is coded as a fixed length chromosome of integer gene indices. The first 12 is randomly sampled. The selection is done with size 3 to achieve a moderate level of selective pressure and diversity. Elite conservation maintains the best 2 to the following generation. Crossover is applied as uniform crossover between chosen parent pairs and adaptive mutation applies perturbations with a base probability of 0.20. Fitness evaluation is used to optimize the candidate-defined residual CNN on a smaller epoch budget and the highest validation accuracy is recorded. The best configuration is used on the last GA-CNN to train it using 80 epochs with cosine learning rate decay and warm restarts after 20 epochs.

D. Jellyfish Search Optimizer

The Jellyfish Search Optimizer represents hyperparameters as continuous real-valued position vectors, where each element of the position vector represents the closest valid discrete candidate value in calculating fitness. The population is made up of 8 jellyfish agents repeated in 6 optimization rounds. A time-control mechanism which is defined as the most important process of JSO:

$$C(t) = \left| \left(1 - \frac{t}{T_{\max}}\right) \times (2 \times \text{rand} - 1) \right|$$

where t is the iteration at hand and Tomax is the overall iterations. At times when C(t) is greater than the constant C = 0.5, the agent is swept away by the ocean current, which is the population mean with a weighted movement towards the global best. In the case where C(t) is below the boundary and a uniform random draw exceeds 1-C(t), the agent makes a passive Levy flight step using the Mantegna algorithm, which permits long-range jumps to get out of local optima. Otherwise, the agent swarms actively move towards or away a randomly selected peer. The proportion of the type of motion was 70.8% active swarming, 16.7% ocean current, and 12.5% Levy flight, which shows gradual change to exploitation as C(t) decays. The best location is then trained on the final JSO-CNN using 60 epochs.

VI. EXPERIMENTAL RESULTS

A. CNN and MobileNetV2 Baseline Results

The Custom CNN gradually converges to 50 epochs, and the training and validation accuracy reaches 0.93 and 0.90 respectively. The training loss decreases to less than 0.30, which proves that the augmentation and regularization approach is effective in avoiding overfitting. The last test result has an accuracy of 90.48 percent and a weighted macro F1-score of 90.40, with automobiles (96.7 percent), frogs (97.6 percent) and trucks (96.2) having the highest per-class accuracy.

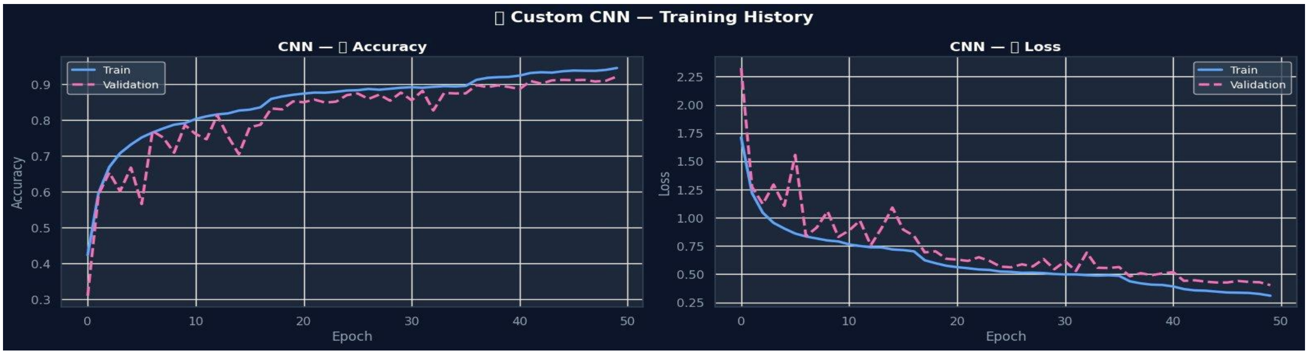


Fig. 2. Custom CNN training and validation accuracy and loss over 50 epochs.

The MobileNetV2 model has a typical Phase 1 plateau in the extraction of features, and the validation accuracy of the model is about 0.80 before the fine-tuning stage starts at epoch 15 as illustrated in Fig. 3. The transition to fine-tuning gives a temporary regression that is seen as the sharp inflection at the dotted fine-tune start marker. Following fine-tuning, the accuracy of validation recovers and approaches 0.87, but the final test accuracy of 82.98% and F1-score of 82.86% are worse than all CNN settings. This poor performance is indicative of the domain difference between ImageNet high-resolution features and the low-resolution 32x32 images of CIFAR-10.

Fig. 3. MobileNetV2 two-phase training history showing accuracy and loss across Phase 1 and fine-tuning.

B. GA-CNN Results

The Genetic Algorithm search is a discrete hyperparameter search which explores the space of hyperparameters in a single generation of 12 individuals, and evaluates each candidate on a 15-epoch proxy training run. The most successful configuration found has a validation accuracy of 89.58% and F1-score of 89.55% on the held-out test set after 80 epochs of cosine annealing training. The training history of the GA-CNN presented in.

Fig. 4 indicates that the first 20 epochs are steep with the highest training accuracy of about 0.91, followed by a temporary regression as a result of the warm restart learning rate cycle at epoch 20. The confusion matrices in Fig. 5 confirm particularly high accuracy on automobiles (96%), horses (94%), ships (94%), and trucks (94%), and cat classification is the most challenging classification (77%).

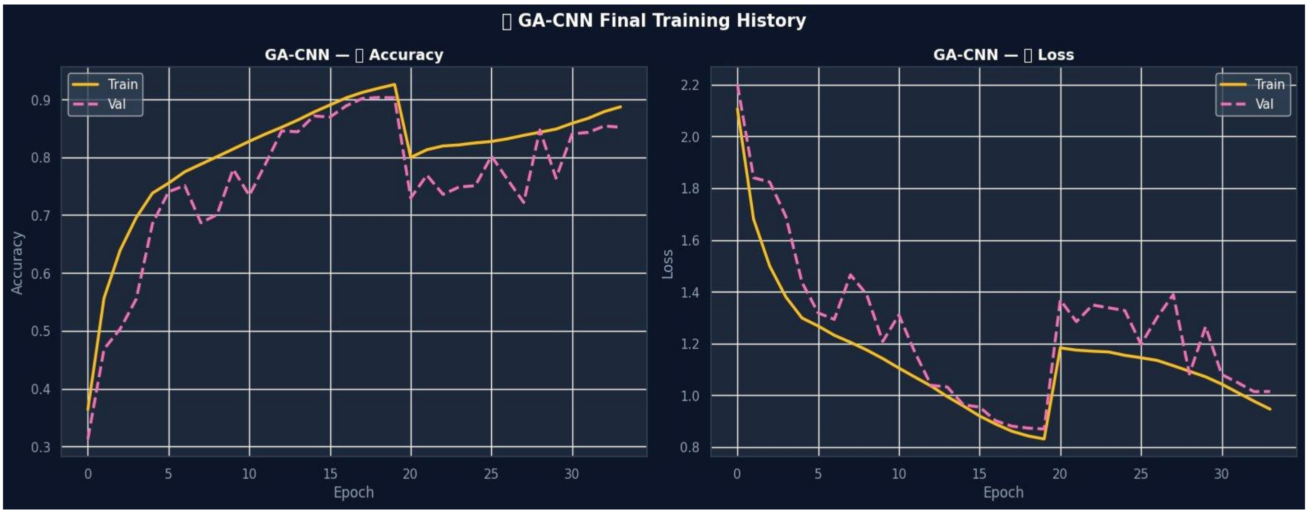
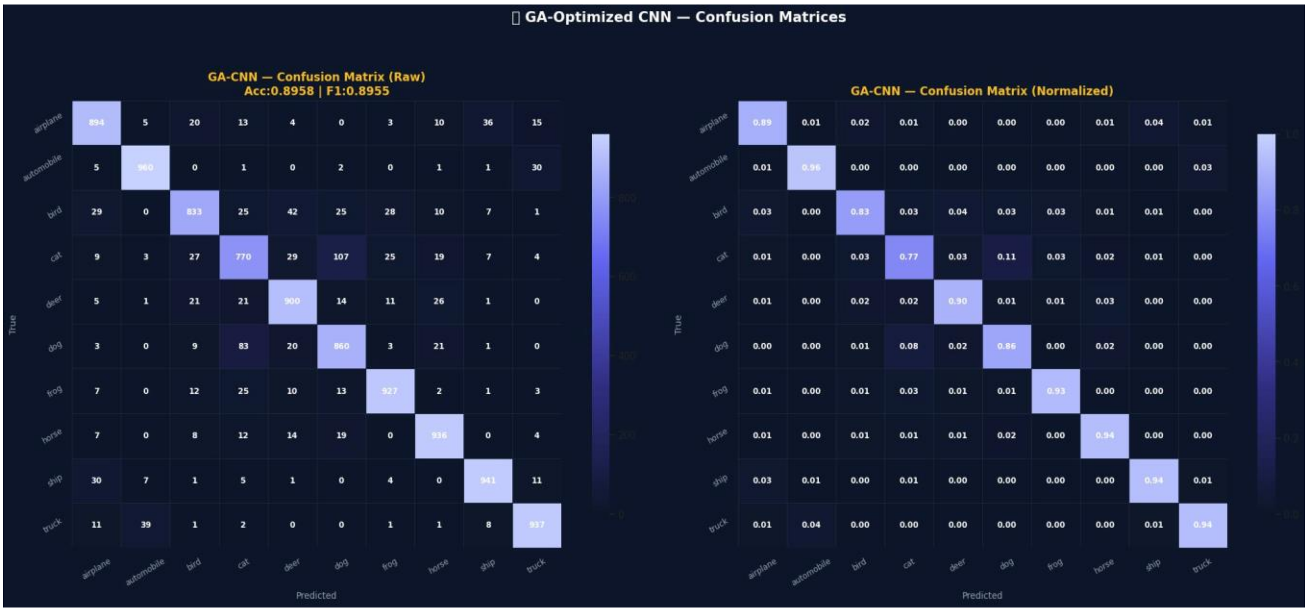


Fig. 4. GA-CNN final training history showing accuracy and loss over 32 epochs with cosine warm restarts.



	airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck
airplane	894	5	20	13	4	0	3	10	36	15
automobile	5	960	0	1	0	2	0	1	1	30
bird	29	0	833	25	42	25	28	10	7	1
cat	9	3	27	770	29	107	25	19	7	4
deer	5	1	21	21	900	14	11	26	1	0
dog	3	0	9	83	20	860	3	21	1	0
frog	7	0	12	25	10	13	927	2	1	3
horse	7	0	8	12	14	19	0	936	0	4
ship	30	7	1	5	1	0	4	0	941	11
truck	11	39	1	2	0	0	1	1	8	937

	airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck
airplane	0.89	0.01	0.02	0.01	0.00	0.00	0.00	0.01	0.04	0.01
automobile	0.01	0.96	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03
bird	0.03	0.00	0.83	0.03	0.04	0.03	0.03	0.01	0.01	0.00
cat	0.01	0.00	0.03	0.77	0.03	0.11	0.03	0.02	0.01	0.00
deer	0.01	0.00	0.02	0.02	0.90	0.01	0.01	0.03	0.00	0.00
dog	0.00	0.00	0.01	0.08	0.02	0.86	0.00	0.02	0.00	0.00
frog	0.01	0.00	0.01	0.03	0.01	0.01	0.93	0.00	0.00	0.00
horse	0.01	0.00	0.01	0.01	0.01	0.02	0.00	0.94	0.00	0.00
ship	0.03	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.94	0.01
truck	0.01	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.94

Fig. 5. GA-Optimized CNN raw and normalized confusion matrices (Acc: 89.58%, F1: 89.55%).

C. JSO-CNN Results

The Jellyfish Search Optimizer has the best overall performance of all the configurations tested. Fig. 6 of the JSO convergence dashboard indicates that the best fitness score is gradually rising over 6 iterations with the best fitness score rising by 0.51 to 0.65 and the average fitness rising by 0.38 to 0.57, which is a progressive swarm concentration to high-fitness regions of the search space. The type of motion pie chart indicates that 70.8 percent of all position changes were active swarming movements, 16.7 percent ocean current drift and 12.5 percent Levy flight jumps.

The hyperparameter configuration that was found after training the final JSO-CNN on 60 epochs has a test accuracy of 91.26% and a macro F1-score of 91.23% as shown in the training history Fig. 7 and confusion matrices Fig. 8. The model achieves the best per-class accuracy on automobiles (97%), frogs (94%), horses (94%), ships (94%), and trucks (94%) and the most remaining confusion is on the classification of birds and dogs, as this is known to be difficult on CIFAR-10 at low resolution.

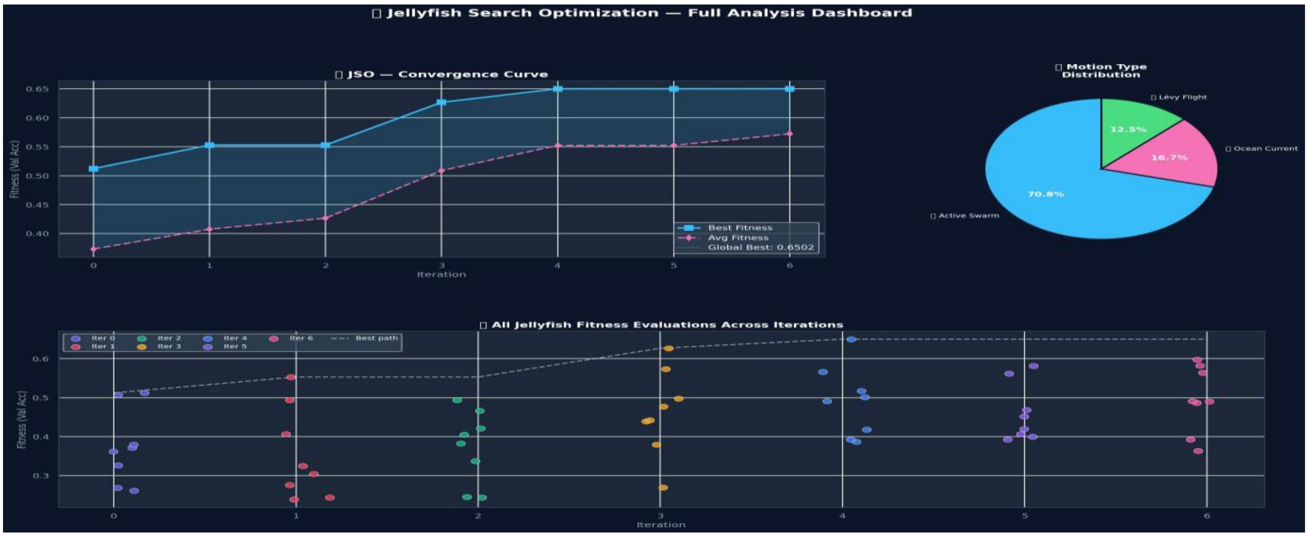


Fig. 6. Jellyfish Search Optimization convergence curve and motion type distribution across 6 iterations.

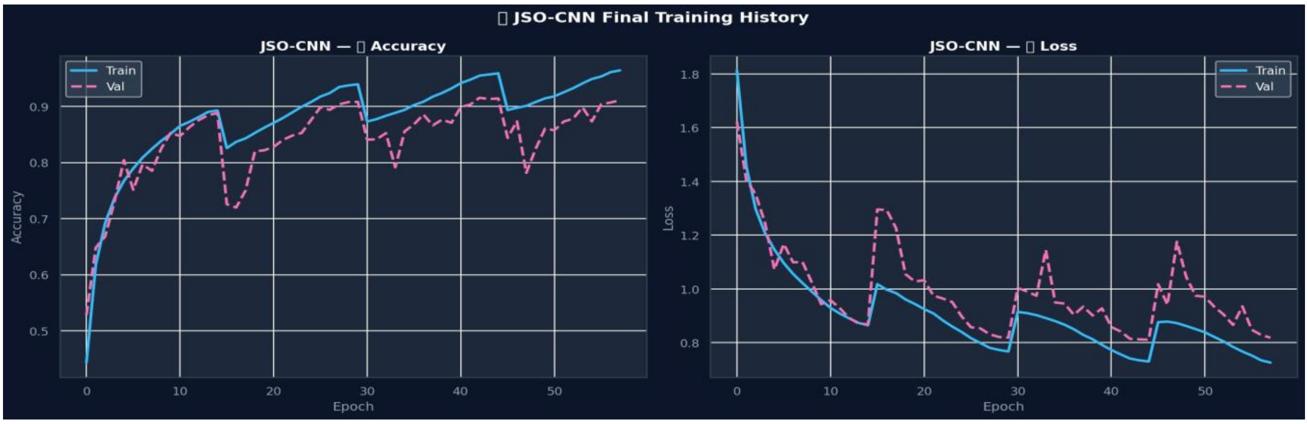
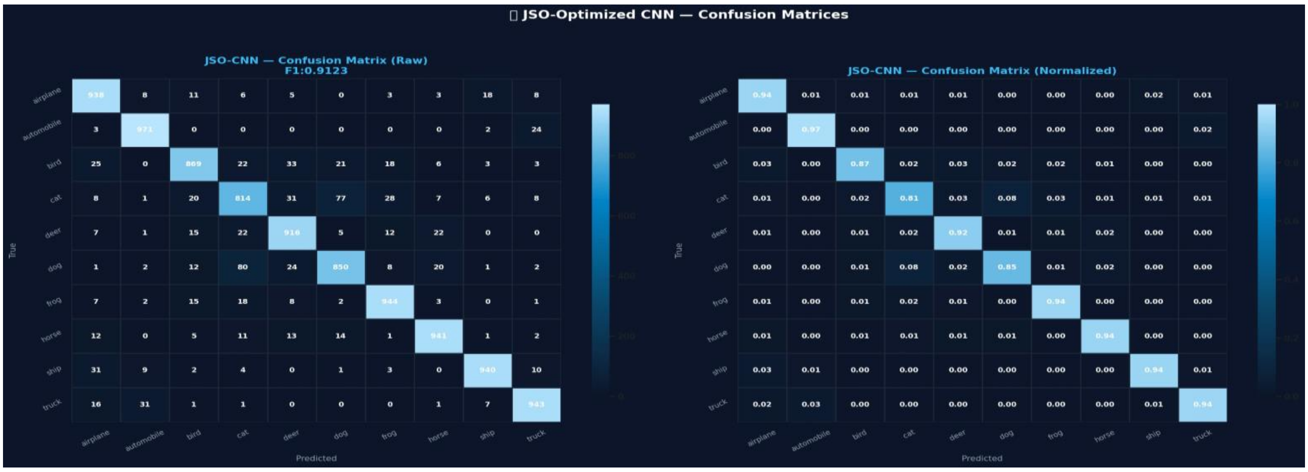


Fig. 7. JSO-CNN final training history: accuracy and loss over 55 epochs.



	airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck
airplane	938	8	11	6	5	0	3	3	18	8
automobile	3	971	0	0	0	0	0	0	2	24
bird	25	0	869	22	33	21	18	6	3	3
cat	8	1	20	814	31	77	28	7	6	8
deer	7	1	15	22	916	5	12	22	0	0
dog	1	2	12	80	24	850	8	20	1	2
frog	7	2	15	18	8	2	944	3	0	1
horse	12	0	5	11	13	14	1	941	1	2
ship	31	9	2	4	0	1	3	0	940	10
truck	16	31	1	1	0	0	0	1	7	943

	airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck
airplane	0.94	0.01	0.01	0.01	0.01	0.00	0.00	0.00	0.02	0.01
automobile	0.00	0.97	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02
bird	0.03	0.00	0.87	0.02	0.03	0.02	0.02	0.01	0.00	0.00
cat	0.01	0.00	0.02	0.81	0.03	0.08	0.03	0.01	0.01	0.01
deer	0.01	0.00	0.01	0.02	0.92	0.01	0.01	0.02	0.00	0.00
dog	0.00	0.00	0.01	0.08	0.02	0.85	0.01	0.02	0.00	0.00
frog	0.01	0.00	0.01	0.02	0.01	0.00	0.94	0.00	0.00	0.00
horse	0.01	0.00	0.01	0.01	0.01	0.01	0.00	0.94	0.00	0.00
ship	0.03	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.94	0.01
truck	0.02	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.94

Fig. 8. JSO-Optimized CNN raw and normalized confusion matrices (Acc: 91.26%, F1: 91.23%).

TABLE I. Comparative Performance Metrics of All Evaluated Models on CIFAR-10

Model	Acc.(%)	Prec.(%)	Rec.(%)	F1(%)
Custom CNN	90.48	90.52	90.48	90.40
MobileNetV2	82.98	83.10	82.98	82.86
GA-CNN	89.58	89.71	89.58	89.55
JSO-CNN (Proposed)	91.26	91.38	91.26	91.23

VII. DISCUSSION

The experimental findings indicate some didactic trends of comparative advantages of manual design, transfer learning, and bio-inspired hyperparameter search to image classification on CIFAR-10. The JSO-CNN performs the best in the four measures, with an accuracy difference of about 0.78 percentage points and F1-score difference of 0.83 points compared to the Custom CNN. This is a small absolute change, but a non-trivial change, which was not obtained by hand experimentation or by injecting domain knowledge into the residual architecture. The reason why

The result of MobileNetV2 is a bit unexpected because the fine-tuned pretrained model is poorer than all CNN settings even though it has much more parameters. This is mainly because the CIFAR-10 images at 32x32 pixels do not have the fine-grained texture and structural detail that depthwise separable convolutions of MobileNetV2 are trained to identify in ImageNet at 224x224. The reconstructed high-frequency detail is still inadequate to drive the pretrained feature extractors to their optimum operating point even after upsampling to 96x96. This observation highlights one of the most common limitations of transfer learning: architectural and preprocessing assumptions in pretrained weights might not be consistent with the statistical properties of the target domain.

The analysis of the confusion matrix in all four models always indicates that cat and dog are the hardest classes, which is a real visual ambiguity between the two classes at low resolution. There are also some models of bird classification that are mixed up with airplane due to the similarity of silhouettes. Practically, the FastAPI-based web application architecture outlined in this paper provides a barebones but functional deployment stack comprising of optimized model artifact to production inference service, authentication layer, user credential management, and asynchronous prediction pipeline.

VIII. CONCLUSION AND FUTURE SCOPE

The paper has introduced a hybrid optimization method to automated hyperparameter optimization of residual Convolutional Neural Networks to classify CIFAR-10 images. Four designs were experimented on equal conditions: a manually designed Custom CNN, a fine-tuned MobileNetV2 transfer learning model, a Genetic Algorithm-optimized CNN, and a Jellyfish Search Optimizer-optimized CNN. The JSO-CNN had the best overall performance with a test accuracy of 91.26% and F1-score of 91.23% showing that the adaptive motion operators of the Jellyfish Search Optimizer allow more efficient exploration of the discrete hyperparameter search space than genetic operators or manual design. The Custom CNN baseline scored 90.48, which proves that a well-regularized architecture with well-selected training schedules is still very competitive. The transfer learning of MobileNetV2 was 82.98 percent, which showed the importance of the correspondence of domains between the pretrained feature statistics and the target image distribution.

Future directions include the JSO search being expanded to the architecture space itself, addition of attention modules to the fitness evaluation objective. The applicability of the suggested framework to other high-stakes domains would be evaluated by implementing it to medical imaging data, such as chest radiograph classification and retinal fundus analysis. The use of federated versions of the optimization loop, in which fitness computations are decentralized to edge devices with access to private image datasets, is a promising direction towards privacy-preserving automated model development.

IX. REFERENCES

[1] E. E.-D. Hemdan, M. A. Shouman, and M. E. Karar, “COVIDX-Net: A Framework of Deep Learning Classifiers to Diagnose COVID-19 in X-Ray Images,” Mar. 2020.

[2] D. Elhani, A. C. Megherbi, A. Zitouni, F. Dornaika, S. Sbaa, and A. Taleb-Ahmed, “Optimizing convolutional neural networks architecture using a modified particle swarm optimization for image classification,” *Expert Syst. Appl.*, vol. 229, Nov. 2023, doi: 10.1016/j.eswa.2023.120411.

[3] M. Kumar, N. Mohd, G. Shivam, A. Goyal, D. Parashar, and R. Khan, “Hybrid Aquila optimizer–Harris Hawks optimization for CNN hyperparameter tuning in brain tumor classification,” *Sci. Rep.*, Mar. 2026, doi: 10.1038/s41598-026-43329-7.

[4] K. Aguerchi, Y. Jabrane, M. Habba, and A. H. El Hassani, “A CNN Hyperparameters Optimization Based on Particle Swarm Optimization for Mammography Breast Cancer Classification,” *J. Imaging*, vol. 10, Feb. 2024, doi: 10.3390/jimaging10020030.

[5] A. Zafar *et al.*, “An Optimization Approach for Convolutional Neural Network Using Non-Dominated Sorted Genetic Algorithm-II,” *Computers, Materials and Continua*, vol. 74, pp. 5641–5661, 2023, doi: 10.32604/cmc.2023.033733.

[6] M. Alnabhan, A. K. Habboush, Q. Abu Al-Haija, A. K. Mohanty, S. Pattnaik, and B. K. Pattanayak, “Hyper-Tuned CNN Using EVO Technique for Efficient Biomedical Image Classification,” *Mobile Information Systems*, vol. 2022, 2022, doi: 10.1155/2022/2123662.

[7] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, International Conference on Learning Representations, ICLR, 2015.

[8] D. Liu, H. Ouyang, S. Li, C. Zhang, and Z. H. Zhan, “Hyperparameters optimization of convolutional neural network based on local autonomous competition harmony search algorithm,” *J. Comput. Des. Eng.*, vol. 10, pp. 1280–1297, Aug. 2023, doi: 10.1093/jcde/qwad050.

[9] T. Y. Hong and C. C. Chen, “Hyperparameter optimization for convolutional neural network by opposite-based particle swarm optimization and an empirical study of photomask defect classification,” *Appl. Soft Comput.*, vol. 148, Nov. 2023, doi: 10.1016/j.asoc.2023.110904.

[10] A. Boulaiche, S. Haddad, and A. Lemouari, “A Convolutional Neural Network with Hyperparameter Tuning for Packet Payload-Based Network Intrusion Detection,” *Symmetry (Basel).*, vol. 16, Sep. 2024, doi: 10.3390/sym16091151.

[11] V. Badrinarayanan, A. Handa, and R. Cipolla, “SegNet: A Deep Convolutional Encoder-Decoder Architecture for Robust Semantic Pixel-Wise Labelling,” May 2015.

[12] S. An, M. Lee, S. Park, H. Yang, and J. So, “An Ensemble of Simple Convolutional Neural Network Models for MNIST Digit Recognition,” Oct. 2020.

[13] Q. S. Hamad, H. Samma, and S. A. Suandi, “Optimization of Convolutional Neural Network Hyperparameter for Medical Image Diagnosis using Metaheuristic Algorithms: A short Recent Review (2019-2022),” Dec. 2024.

[14] K. H. Abdalla, F. Daneshfar, E. Hosseini, and M. Oussalah, “HyOCNN: Hybrid-optimized convolutional neural network for robust image classification,”

Biomed. Signal Process. Control, vol. 113, Mar. 2026, doi: 10.1016/j.bspc.2025.109040.

[15] S. Bakas *et al.*, "Identifying the Best Machine Learning Algorithms for Brain Tumor Segmentation, Progression Assessment, and Overall Survival Prediction in the BRATS Challenge," Apr. 2019.