Optimization Approach: Classification of Brain Tumor Images Using Convolutional Neural Network (EfficientNet)

Alberto Wicker,
MSc Medical Informatics,
University of Applied Sciences and
Arts Northwestern Switzerland,
Muttenz, Switzerland
alberto.wickervera@students.fhnw.ch

Thomas Hanne,
Institute for Information Systems,
University of Applied Sciences and
Arts Northwestern Switzerland,
Olten, Switzerland
thomas.hanne@fhnw.ch

Rolf Dornberger,
Institute for Information Systems,
University of Applied Sciences and
Arts Northwestern Switzerland,
Basel, Switzerland
rolf.dornberger@fhnw.ch

***Abstract*— This study investigates medical image classification employing Convolutional Neural Networks (CNNs), Support Vector Machine (SVM and Genetic Algorithm (GA); focusing on hyperparameter optimization for the task of Brain Tumor diagnosis with Magnetic Resonance Imaging (MRI) scans. A brief discussion of other CNN applications in other diagnosis scenarios is done in: Diagnosis of Gastrointestinal disorders using endoscopy and Blepharoptosis identification. Different data categories and attributes determine the most effective model for categorizing images correctly according to pre-established classification features. The aim is to employ a hybrid CNN-SVM model to facilitate the exhaustive task of multi-class classification and determine optimal hyperparameters (Genetic Algorithm) suitable for the diagnostic task. This paper analyzes critical factors affecting CNN performance, focusing on parameters that substantially influence the model effectiveness. The models will be compared as standalone and combined (all together) to contribute to the knowledge of CNN applications in image analysis in the medical radiological field.**

***Keywords*—** *Convolutional Neural Networks, Medical Image Classification, Diagnostic Imaging, Deep Learning, Machine Learning, Image Processing, Support Vector Machines, Genetic Algorithm.*

I. Introduction

Almost 90% of all data in healthcare is comprised of medical images [1] and recent advancements in Convolutional Neural Networks (CNNs) have impacted the field of medical image classification by introducing enhancing techniques for a wide range of diagnosis tasks. The architecture of CNNs can significantly differ based on the specific application within medical image analysis. Unlike traditional methods that demand extensive feature selection, CNN models can automate this process, contributing to an upgrade in performance [2]. This analysis explores the transformative role of CNNs across multiple diagnostic challenges emphasizing their ability to improve image diagnosis accuracy in medical imaging.

Furthermore, the paper examines how extracting valuable features and complexities within CNN architectures influences diagnostic accuracy. It provides comprehensive insights into the methodologies applied in these distinct image-based diagnostic tasks to achieve optimal results in anomaly identification and categorization for a wide range of imaging diagnosis tasks [3]. The objective is to investigate more accurate, adaptable, and automated methods for the task of classifying 3 classes of Brain Tumour MRI images: Glioma, Meningioma, Pituitary tumour. The ultimate goal is to enhance diagnostic accuracy and optimize a baseline CNN model for medical image classification.

II. Description of the Optimization Problem

Medical imaging scenarios focus on reinforcing image classification accuracy. Variables in common include image resolution, model architecture, and image pre-processing techniques. Nevertheless, the specific medical condition and CNN methods diverge. Eleven CNN models were employed when identifying blepharoptosis, and DenseNet121 architecture without pre-training achieved the best results (sensitivity: 90.1 % and specificity: 82.4 %) [4]. In the exploration of gastrointestinal disease diagnosis, various CNN models were applied. Some models started from scratch, while others used transfer learning techniques. For instance, the ResNet50 model, which utilized transfer learning, was specifically trained on the KVASIR multi-class image dataset and achieved an accuracy of 99%. [5]. This diagnostic case demonstrates the adaptability of diverse CNN architectures in handling specialized types of medical images while also demonstrating the burden of highly personalized approaches.

A. Problem Model

Using transfer learning models consisting of CNNs and Support Vector Machines (SVM) can mitigate the limitations of the traditional approach, as suggested by I. Bouslihim et al. Initially the features are extracted by the neural network, and then multi-class classification is performed by the SVM mapping CNN features in a high dimensional space. To achieve the goal of designing a model that can be optimized for correctly classifying images, this paper conducts an analysis of a deep learning and machine learning methods.

CNN's initial layers detect essential input elements, while later layers merge these to identify complex features. Filters in convolutional layers scan the input to create feature maps, highlighting patterns (e.g., edges). Pooling layers then reduce size for efficiency and to avoid overfitting. Finally, fully connected layers make predictions, like categorizing images. The basic layer architecture of CNN is shown in Figure 1.

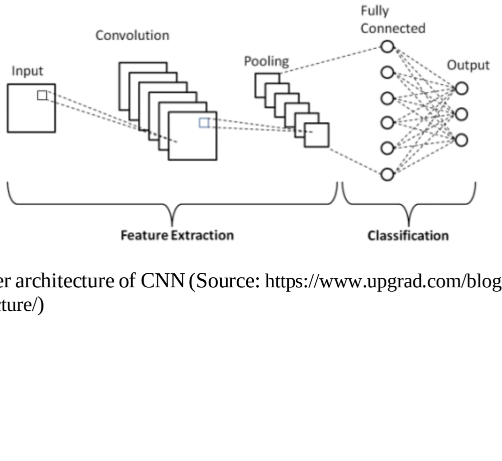

Fig. 1. Layer architecture of CNN (Source: https://www.upgrad.com/blog/basic-cnn-architecture/)

The SVM is a robust classifier adept for linear and non-linear challenges. It applies kernel functions and separates data into classes using a line or hyperplane, as shown in Figure 2. This allows for the classes to be mapped in a high dimensional space separated by the stemming boundary. The main disadvantage is careful hyperparameter tuning and specific pre-processing for medical images [6]. Creating a compound model alleviates the main pitfalls of CNN and SVM.

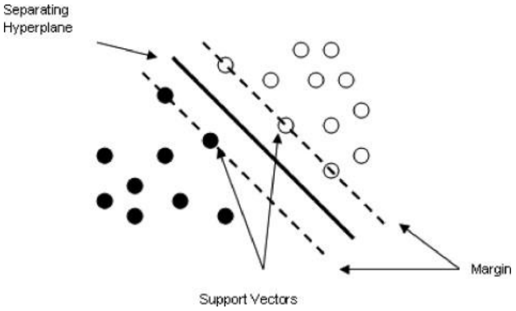

Fig. 2. SVM separating hyperparameters and margins (Source: https://www.sciencedirect.com/topics/computer-science/separating-hyperplane)

B. *Optimization Methods*

The main component in the approach the present classification problem is EfficientNetB0 a small, yet efficient CNN that is the base for the EfficientNet family (B0-B7). The architecture of this models allows for compound scaling [7], this mechanism consistently balances the network in three dimensions: width, depth and resolution; this harmony is effectuated through the application of a scaling constant ratio with a fixed set of coefficients provided in Figure 3.

$$\begin{aligned}
\text{depth: } & d = \alpha^{\phi} \\
\text{width: } & w = \beta^{\phi} \\
\text{resolution: } & r = \gamma^{\phi} \\
\text{s.t. } & \alpha \cdot \beta^2 \cdot \gamma^2 \approx 2 \\
& \alpha \geq 1, \beta \geq 1, \gamma \geq 1
\end{aligned}$$

Fig. 3. EfficientNet balanced scaling method with compound coefficient (Source: https://viso.ai/deep-learning/efficientnet/)

A grid search algorithm determines the values of α, β and γ, the overall scaling factor is designated by the parameter ϕ. Computational resources are dictated by the constraining equation so that scaling the network increases the floating point operations per second (FLOPS) by approximately 2:

1) *EfficientNet Scaling:*

$$(\alpha \cdot \beta^2 \cdot \gamma^2)^{\phi}$$

Hence the total FLOPS increase by 2^{ϕ} . In the case of EfficientNetB0 ϕ is configured to 1, indicating twice the resources, the generated values maintain the constraining equation.

The search in EfficientNet models B1-B7 derives from scaling the network with different ϕ values, making the computational costs escalate reasonably.

EfficientNet addresses the limitations of conventional CNN models, often leading to extend architecture design times. This model ensures that better performance with few parameters and lower computational burden is executed by balancing dimensions with uniformly applied values across all layers of the network, M. Tan et al. In an initial stage of this research the EfficientNet model B0 will carry out the classification task of the brain tumour images. The CNN model, already pre-trained from the first task with an extensive dataset provides the foundation for the later stages with intricate extracted features.

Despite varied applications, the primary optimization goals remain constant: enhancing diagnostic accuracy and reducing errors. CNN pre-trained models offer time savings, while custom models often yield superior performance particularly with small or variable datasets, such as in blepharoptosis identification case of Ju-Yi Hung et al. The choice of CNN architecture influences the performance, therefore a careful consideration based on task-specific requirements is required. A common limitation is the need for extensive and homogeneous training datasets, leading to overfitting and hampering the models' generalization ability as discussed by S.S Yadav et al.

Secondly, as previously outlined, SVM will take-on the task of performing an additional classification by utilizing the complex features extracted by the CNN. In a manner of succession and by facilitating separable classes, it is expected that the classification results are enhanced by the SVM.

Essential to the SVM classification is the selection of appropriate hyperparameters to upgrade the performance (discussed in the methodology section). This involves a systematic search for the optimal parameters to find the ideal combination that returns the highest classification accuracy possible. Integrating grid search promotes the refinement of the parameter selection by evaluating different combinations of hyperparameters to carefully identify the set with the best performance [8]. Additionally this approach guarantees dimensionality reduction of the extracted features before using them as input in the SVM and that the final hyperparameter set generalizes properly on unseen data.

Finally, a Genetic Algorithm (GA) partakes in a concluding phase to optimize the set of hyperparameters used in the previous SVM classification. GAs are a subset of machine learning (ML) models for optimization, their conduct derives from replicating processes in evolutive nature (e.g. natural selection and genetics). The GA doesn't directly deal with the values of the parameters but rather a coded form, the exploration doesn't start from a single element but rather the population with deterministic rules for crossover and mutation [9].

The GA is established with an initial population is created, in this case the number of potential solutions and they are represented by chromosomes (set of SVM hyperparameters); Within this context the classification accuracy will serve as a fitness function to evaluate each entity in the population. The fundamental structure for the GA is represented in Figure 4 and includes the following [10]:

1) Starts with an initial population (P_1) that is generated from random N solutions, then the fitness of the solutions in P_1 is evaluated.

2) Matting or Crossover is based on fitness values of two solutions (x and y) from the population (P_t), offspring will be generated and added to a new population (Q_t).

3) The mutation of individual solutions ($x\varepsilon Q_t$) occurs by predefining by a mutation rate.

4) A fitness value is assigned to each offspring ($x\varepsilon Q_t$) conditional to its function value and unlikelihood.

5) N solutions are selected from Q_t and copied to a new population (P_{t+1}) conditional to their fitness.

6) The search stops when the termination criteria are met, otherwise the mating and subsequent processes are repeated.

The GA's artificial natural selection approach prioritizes candidates with superior fitness scores, ensuring that optimal set of traits (in this context SVM hyperparameters), are inherited by the next generation. Offspring inherit traits from both parents from crossover, akin to DNA recombination. Random changes are induced by mutations to maintain a diverse population with the end of precluding premature termination on suboptimal solutions. To yield the optimal hyperparameters the progressive refinement of parameters occurs from iterating over multiple generations.

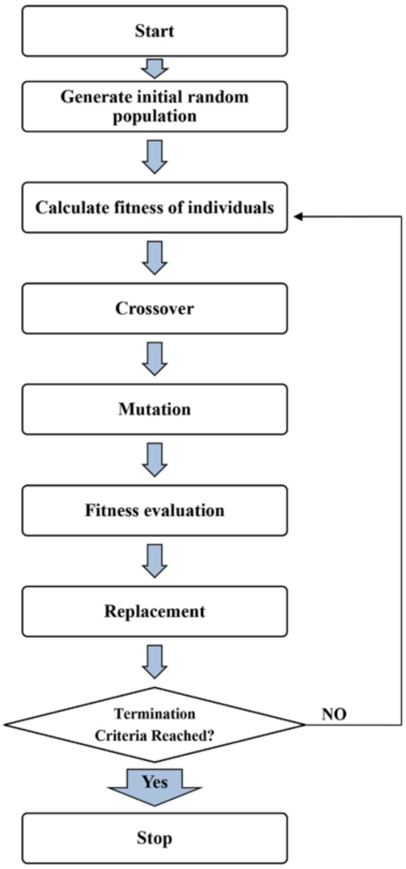
Start
Generate initial random population
Calculate fitness of individuals
Crossover
Mutation
Fitness evaluation
Replacement
Termination Criteria Reached?
NO
Yes
Stop

Fig. 4. Structure Genetic Algorithm (Source: 10.1080/00038628.2018.1477043)

C. *Implementation and Comparison of Different Optimization Methods.*

The employed dataset was sourced from Kaggle [11] encompasses 7,023 MRI images for three different classes of brain tumors (glioma, meningioma and pituitary tumor) and includes MRI images of normal brain scans (no tumor). The training and testing sets consists of 5,712 and 1,311 images respectively. Neuroimaging samples of the characteristics of the brain tumor classes is provided in Figure 5.

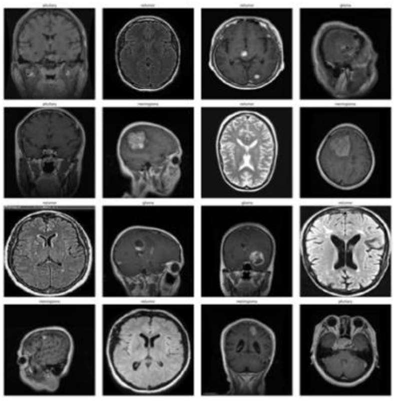

Fig. 5. Diagnostic MRI image samples for Brain Tumours and healthy samples (Image Source: 10.34740/kaggle/dsv/2645886)

This section discusses the employed methodology for constructing an EfficientNet model for image classification and transfer learning with Support Vector Machine and Genetic Algorithm, as proposed by Bouslihim et al. and Z. Szymanski. Results are discussed accordingly in its respective section.

EfficientNet:

The EfficientNetB0 model is employed for the classification of brain tumour images and as feature extractor in later phases. The set up for this phase was performed in Google Colab coding platform. The construction of the model initiates with loading the datasets (training and testing) and pre-processing. Images are resized to 224x224 pixels using the 'preprocess_input' function from TensorFlow's EfficientNet module to ensure a suitable pixel range for the model. 'ImageDataGenerator' is used to augment the data normalization, configured to a batch size of 16 and an image size of 224x224 pixels. This generator applies random changes to the data to increase diversity among the training set, aiding in the model's ability to generalize. A validation set is created from the testing dataset by using a split and securing an invariable division in each run with 'random_state'.

The model is initialized with pre-trained weights from 'ImageNet'. The architecture is assembled with 'TensorFlow' and 'Keras', encompassing several layers, including: Batch Normalization, Flatten to convert the matrices into an individual vector, Dense layer with 256 neurons in unit size, ReLu (activation parameter), Dropout layer to reduce the probability of overfitting with a rate of 0.5, and a final Dense layer for multi-class classification with 'soft_max' activation function.

The model is compiled applying loss and accuracy as the metric with 'Adam' optimizer set up with a learning rate of 0.0001, and configured for categorical cross entropy loss function optimization, with 'accuracy' as primary metric to evaluate performance during training. The training comprised of running the model for three epochs and tracking the performance with loss and accuracy metrics on the training and validation sets (Figure 6). Finally, The model is saved for further use.

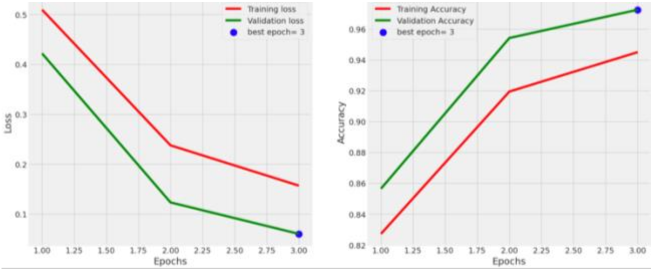

Fig. 6. Loss and Accuracy performance over epochs in the EfficientNetB0 model

SVM for Classification Optimization:

SVM is employed as an alternate classifier using extracted features from the saved CNN model, this phase was performed in Google Colab. Initially the images are resized to 224x224 pixels for consistency. The EfficientNetB0 model is loaded to engage as feature extractor, by retaining the layers up to the flatten layer the output size of feature vectors is of 1280. The features are reconstituted into a one dimension input to serve as input for the SVM training. 'StandardScaler' from scikit-learn was employed to guarantee feature vectors with a standard deviation of one and a mean of zero.

The SVM was set up initiated with a Radial Basis Function (RBF) kernel to handle non-linear relationships within the features. 'GridSeearchCV' was used to test multiple combinations of the regularization parameter (*C*) and the kernel coefficient (*gamma*) to identify the best parameters of the SVM. The grid search is done with a five-fold cross-validation to ensure the selected hyperparameters generalize well on unseen data. The SVM is trained on the feature vectors using the established parameters. During the training process 'parallel_backend' from joblib expedited the computation by enabling lateral processing. Table I summarizes the key aspects of the SVM implementation phase.

TABLE I. SVM GRID SEARCH CONFIGURATION

Parameter	***Description***
Kernel	Radial Basis Function
Parameters	C: [0.1, 1, 10], gamma: ['scale', 0.001, 0.01]
Total Fits	45 (9 parameter combinations, 5 folds for each)

Genetic Algorithm for SVM Hyperparameter Optimization:

To optimize the SVM hyperparameters, a GA was employed. 'PyGAD' library was used to establish the GA in Python coding language, employing Visual Studio Code (VSC). The purpose of this phase is to optimize the previously used SVM hyperparameters to appraise if there is a noticeable superiority in accuracy through GA in search for a fitter model configuration.

Initially the generation of a population with random values within a specified range for C and gamma was done accordingly to the previously handled parameters and the population was set to 10. Each solution is represented by a pair of values for C and gamma. The population size is set with each solution represented by a pair of values for C and gamma:

- C:0.1 to 10.0
- gamma: 0.0001 to 1.0

To evaluate performance of each solution, the fitness value was defined as the SVM's accuracy on the test dataset, extracted using the EfficientNetB0. To evaluate performance of each individual in the population by training the SVM with corresponding C and gamma values, fitness function was employed. The fitness value was defined as the accuracy of the SVM on the test dataset, which was extracted using the EfficientNetB0. The mutation is established at a rate of 50% to secure that at least one gene is mutated, only the top individuals are selected to participate in cross-over. The best solutions and fitness values are tracked along 30 generations. The `on_generation` callback function was used to log and visualize the progress of the GA. Table II summarizes the configuration.

TABLE II. GENETIC ALGORYTHM OPERATORS

Parameter	***Description***
No. Generations	30
Populationn	10
Selection	Top 5 individuals
Crossover	Single Point
Mutation Rate	50%
Gene Space	Kernel: RBF, C: [0.1 - 10], gamma: [0.001 - 0.01]

After iterating and refining over the generations, the performance was visualized using 'Plotly' showing the best fitness values in a timeline. The best hyperparameters in the final generation are selected as the optimized set for SVM classification.

III. RESULTS

To identify the best classifying performance and establish a comparison among models the following performance metrics are employed and defined below [5,12]:

1) Accuracy: Measures the correctly classified cases amonng the total cases. Provides a comprehensive effectivenes of the classifier.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

2) Precision: Proportion of True Positive predictions whithin all the Positive predicitons. A higher value respresents less occurrence of false possitives.

$$Precission = \frac{TP}{TP + FP}$$

3) Recall: Classifier ability to correctly identify relevant cases. A higher value indicates less false negatives.

$$Recall = \frac{TP}{TP + FN}$$

4) F1-Score: Harmonic mean between precision and recall. Useful for a thorough performance evaluation.

$$F1 - Score = 2 \times \frac{(Recall)(Precision)}{Recall + Precision}$$

Table III summarizes the performance results achieved by each of the evaluated models in the test sets.

TABLE III. Performance Results Summary.

Model	***Performance Metrics Overall (%)***			
	Accuracy	***Precision***	***Recall***	***F1-Score***
CNN	98.02	98.00	98.00	98.00
CNN-SVM	98.47	98.32	98.27	98.31
CNN-SVM-GA	98.17	98.08	98.17	98.15

The base model (EfficientNetB0) obtained accuracy of 97.25% in the validation set, 98.02% in the test set and 98.02% as an overall accuracy. Using the features of the CNN model the SVM obtained accomplished a higher overall accuracy with 98.47%. Finally in the GA optimization using SVM hyperparameters, the best solution after 30 generations yielded a lower overall accuracy of 98.17%, a slight improvement from the baseline model, although lower in comparison to the SVM standalone classification (Figure 7).

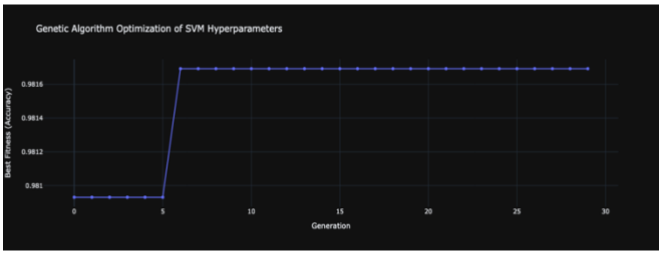

Fig. 7. Fitness values over Genetic Algorithm generations

As expressed by A. Tanner et al. a confusion matrix (CM) is employed to graphically convey classification accuracy. This method is commonly used in machine learning being that contains information about the class prediction. Figure 8 illustrates the performance in classifying each class: glioma, meningioma, no tumor and pituitary. The CM for EfficientNetB0's conveys that misclassifications were rare, indicating promise in characterizing between tumor classes and patients with no tumor in the validation set. The SVM achieved perfect classification for no tumor scans. This model correctly classified 292 glioma, 295 meningioma, 405 no tumor, and 299 pituitary images, with only 20 misclassifications inn the test set. The optimization of SVM hyperparameters with GA demonstrated appropriate classification with only 24 missed cases.

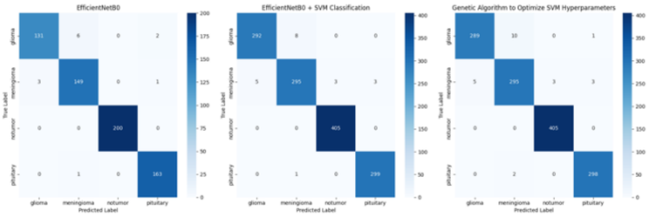

Fig. 8. Confusion Matrices (left to right): EfficientNetB0, SVM classification, SVM with GA hyperparameter Optimization

The final workflow showcases the potential the integration of deep learning models with deep learning techniques for image classification tasks. The EfficientNetB0 model provided a decent groundwork, the classification with SVM classifier improved overall performance metrics and the GA, offers channels for further optimization.

Future work could test models within the EfficientNet family, B0 to B7 to inquire on models that may deliver better results and implement them in the optimization pipeline. Additionally, SVM as a final classifier, presented improved results although with improved computational resources an extended list of kernels and parameters could be tested in order to expand the search and improve performance. Genetic algorithm proved to be less effective, but could benefit from increased population size and an extended generation line to optimize hyperparameters. Such adjustments could potentially lead to better performance outcomes. The correlation of results in this pipeline can impact the next phase, therefore integrating a magnified range for EfficientNet models and parameters in the next phases can be of great use. Finally employing a larger dataset or testing this workflow in other medical imaging diagnostic tasks could also open a new window for research.

IV. Conclusions

This study searches for the potential advantages and shortcomings of optimizing EfficientNetB0 model for classifying brain tumour MRI scans by using SVM as the final classifier. It also examines how a Genetic Algorithm could help optimize SVM's hyperparameters to improve accuracy. In this scenario, the EfficientNetB0 served as a foundation for feature extraction, the SVM proved to enhance classification capabilities and the GA was used to optimize the SVM hyperparameters yet it showed marginal improvements compared to the SVM alone. EfficientNetB0 model achieved an overall accuracy of 98.02%. The CNN-SVM classification improved the overall accuracy to 98.47% and CNN-SVM-GA model achieved an accuracy of 98.17%. The confusion matrices show that misclassifications are rare, specially in the SVM model. This study represents the possibility of combining deep learning models with optimization techniques in the case of medical image classification. While not showing any improvement, the GA introduces an avenue for further optimisation via larger populations and generations. In future work exploration of the EfficientNet Family models to identify better performance and integrate them into this optimization workflow. Additionally, extending the configuration of SVM parameters and testing this workflow on larger data or other medical tasks could provide new insights and applications.

REFERENCES

[1] G. Papanastasiou, “Focus on machine learning models in medical imaging,” Physics in Medicine &amp; Biology, vol. 68, no. 1, p. 010301, Dec. 2022. doi:10.1088/1361-6560/aca069.

[2] S. S. Yadav and S. M. Jadhav, "Deep convolutional neural network based medical image classification for disease diagnosis," Journal of Big Data, vol. 6, no. 113, 2019. doi: 10.1186/s40537-019-0276-2.

[3] I. Bouslihim, W. Cherif, and M. Kissi, “Application of a hybrid EfficientNet-SVM model to medical image classification,” 2023 14th International Conference on Intelligent Systems: Theories and Applications (SITA), Nov. 2023. doi:10.1109/sita60746.2023.10373755.

[4] Ju-Yi Hung, C. Perera, Ke-Wei Chen, "A deep learning approach to identify blepharoptosis by convolutional neural networks," 2021 International Journal of Medical Informatics, 148 (2021) 104402, pp. 1-7, doi: 10.1016/j.ijmedinf.2021.104402.

[5] A. Sharma, R. Kumar, P. Garg, " Deep learning-based prediction model for diagnosing gastrointestinal diseases using endoscopy images," 2023 International Journal of Medical Informatics, 177 (2023) 105142, pp. 1-10, doi: 10.1016/j.ijmedinf.2023.105142.

[6] N. J. Sairamya et al., "Hybrid approach for classification of electroencephalographic signals using time–frequency images with wavelets and texture features," in Intelligent Data Analysis for Biomedical Applications, 2019, pp. 253–273, doi: 10.1016/b978-0-12-815553-0.00013-6.

[7] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," International Conference on Machine Learning, 2020, [Preprint]. doi: 10.48550/arXiv.1905.11946.

[8] M. Nashaat, "Hyperparameter Tuning with GridSearchCV," Medium, 22 Oct. 2023. [Online]. Available: medium.com/@mohammednashaat29/hyperparameter-tuning-with-gridsearchcv-8724f215a383.

[9] Z. Szymanski, S. Jankowski, and D. Grelow, "Optimization of support vector machine hyperparameters by using genetic algorithm," in SPIE Proceedings, vol. 6159, 2006. doi: 10.1117/12.674867.

[10] S. Mirniazmandan et al., "Mutual effect of geometric modifications and diagrid structure on structural optimization of Tall Buildings," Architectural Science Review, vol. 61, no. 6, pp. 371-383, 2018, doi: 10.1080/00038628.2018.1477043.

[11] M. Nickparvar, "Brain tumor MRI dataset," [Data set], Kaggle, K. M. Schmainda, M. Prah, "Data from Brain-Tumor-Progression," The Cancer Imaging Archive, 2018. doi: 10.34740/KAGGLE/DSV/2645886. Available: https://doi.org/10.7937/K9/TCIA.2018.15quzvnb

[12] A. Taner, Y. B. Öztekin, and H. Duran, “Performance analysis of deep learning CNN models for variety classification in hazelnut,” Sustainability, vol. 13, no. 12, p. 6527, Jun. 2021. doi:10.3390/su13126527