

Automated Cancer Detection Using CNNs: Innovations and Applications

Ruijie Huang

Faculty of Science and Technology, Beijing Normal University-Hong Kong Baptist University United International College, Zhuhai, 519087, China

s230034077@mail.uic.edu.cn

Abstract. Large-scale datasets and complicated cell architectures provide challenges for traditional tumor identification approaches, which often rely on human analysis or traditional machine-learning techniques. This may result in errors and inefficiencies. The inconsistent shape of tumor cells may be a challenge for these methods, making border detection and classification inconsistent. This paper suggests an automated tumor detection technique based on convolutional neural networks (CNNs) to increase tumor cell border detection and classification accuracy and efficiency. Several experiments were carried out to confirm the suggested strategy's efficacy. This entails developing and refining the CNN architecture in addition to merging deep learning and data augmentation methods. According to the findings, the model has performed well in a variety of experimental settings, especially when it comes to classifying complicated cell samples. Furthermore, the feasibility of using this model for real-time border detection was explored, indicating its suitability for use in a medical setting. Subsequent investigations will concentrate on enhancing the capabilities of models, expanding their range, and tackling problems like disparities in data. This discovery is significant because it has the potential to transform tumor identification by providing a more accurate, scalable, and efficient approach that might be used extensively in clinical settings and eventually lead to better patient outcomes.

Keywords: Automated cancer detection, CNN, Deep Learning, Medical imaging, Image processing.

1. Introduction

A patient's chance of survival is greatly increased by early cancer identification. Early cancer detection frequently allows for intervention before the disease progresses, greatly enhancing the efficacy of treatment. Nevertheless, the interpretation of medical images by skilled physicians is the foundation of conventional cancer detection techniques, which are not only arduous and time-consuming but also subject to human error. Deep learning, and convolutional neural networks in particular, have demonstrated significant promise in medical image processing in recent years. These technologies can perform better than human specialists in many picture classification jobs because they can automatically extract features from images and learn from vast volumes of data. Nevertheless, even though current automated cancer detection techniques have somewhat increased detection efficiency, they still have issues with data [1].

This study aims to design and optimize an efficient CNN model for the automatic detection of cancer. By combining advanced deep learning techniques, the aim is to develop a model that excels in accuracy, robustness, and computational efficiency. Innovations in this study include the use of data enhancement techniques to improve the model's adaptability to unbalanced data, as well as the model's ability to recognize different types of cancer lesions through multi-level feature extraction mechanisms. Additionally, the application prospects of the model in real-world clinical settings are explored, with an evaluation of its generality and scalability across different imaging devices, patient populations, and cancer types. This study contributes to the advancement of automated cancer detection technology and offers a novel approach for the broad application of intelligent medical treatment [2].

2. Methodology

2.1. Dataset Preparation

2.1.1 Data set source

PanopTILs is a comprehensive, large-scale dataset featuring annotated boundaries of histologic regions, along with the classification labels, segmentation boundaries, and locations of nuclei within the same fields of view. This dataset was developed by merging and enhancing two existing public datasets, BCSS and NuCLS, to facilitate an in-depth investigation of the tumor microenvironment in whole-slide images (WSI) of H&E-stained breast cancer slides. Panoptic segmentation is particularly beneficial for evaluating tumor-infiltrating lymphocytes (TILs) in line with clinical guidelines. The integration of this dataset allows for the training of deep learning models that yield biologically relevant predictions [3].

2.1.2 Description of the data set

The dataset consists of regions of interest (ROIs) measuring 1024x1024 pixels, captured at a resolution of 0.25 microns-per-pixel (MPP), equivalent to 40x magnification on various scanners. It provides annotations for histologic regions (semantic segmentation) alongside nuclear classifications and segmentations (object segmentation) for the same ROI, enabling thorough detection of all pertinent components within the tumor microenvironment in H&E scanned slides. These labels characterize different types of nuclei and background information in microscopic images, including cancer cell nuclei, stromal nuclei, large stromal nuclei, lymphoid nuclei, plasma cells or large tumor-infiltrating lymphoid nuclei, normal epithelial nuclei, and other unknown or obscure cell types. The dataset also identifies areas to be excluded and background material devoid of nuclei. These classifications enhance the accuracy of tissue analysis and pathology research [4].

2.1.3 Image normalization and standardization

In the data preprocessing process, the label data is first loaded and processed from multiple CSV files. These tags represent different nuclear types, which are converted to numerical form by 'LabelEncoder' and further converted to unique thermal encoding by 'OneHotEncoder'. This processing allows labels to be used for multiple classification tasks, ensuring that the model is trained to correctly identify and classify different types of nuclei. In addition, the coordinate information of the image region (such as 'left', 'right', 'top', 'bottom') is extracted from the CSV file. These coordinates are used to accurately locate the region of interest in the image so that the model can focus on the key parts for analysis.

In the aspect of image preprocessing, the image and mask are first loaded through the 'load_img' function, and all the images and masks are adjusted to the uniform size '(224, 224)'. This dimensional standardization not only improves computational efficiency but also ensures the consistency of input data. After the images are loaded, they are converted to array form by the 'img_to_array' function, and all images are normalized to scale the pixel values between 0 and 1. The purpose of normalization is to reduce the scale difference of input data, to avoid the training instability caused by different pixel value ranges, and to speed up the convergence of the model.

The mask image is processed similarly to the image, but with grayscale mode loading and special scaling processing - the mask pixel value is divided by 8.0 and then rounded by the 'np. round' function. This operation ensures that the pixel values in the mask are integers, clearly representing different classes of nuclei or background regions, such as cancer cell nuclei, interstitial nuclei, etc. These masks play an important role in model training, marking the categories of regions in the image and helping the model learn how to distinguish between different types of nuclei.

Throughout the process, each pair of images and masks is matched with the corresponding label. If an image or mask file is missing, the system skips the file and issues a warning, thus ensuring the integrity and consistency of the dataset. Valid images, masks, and labels are eventually stored in the 'images', 'masks', and 'valid_labels' lists, which will be used for subsequent model training.

Manipulation of the coordinate array is another key step, the coordinates extracted from the CSV file are stored in the array and are intercepted in the code ('coordinates =coordinates [:745, :]'), probably because there are only 745 images in the data set. Or the data is representative of the entire data set. These coordinates are used for further processing, such as cropping or labeling specific areas in the image, ensuring that the model can focus on the most relevant parts for training.

Finally, masks are applied to the image ('X = images * masks') via dot multiplication operations, which effectively mask irrelevant areas of the image, leaving only the parts of interest. The result is an array of images with masks that will be used to train the model, helping it more accurately identify and classify different types of nuclei when processing real data [5].

Through these elaborate pre-processing steps, the quality of the input data is significantly improved. Label processing enables the model to correctly understand the classification task, and the standardization and normalization of image and mask processing ensures the stability and efficiency of model training. Ultimately, the goal of all these preprocessing operations is to improve the training effect of the model, so that it can achieve higher prediction accuracy and better generalization ability in practical applications.

2.2. Model Architecture and Justification

2.2.1 Introduction to the classic CNN model

CNNs are a kind of deep learning architecture that is effective in solving image processing and computer vision challenges. CNN's primary purpose is to employ the fully connected layer for classification or regression and the convolutional layer for extracting local information from the image. Weight sharing and local connection capabilities allow CNN to drastically reduce the number of parameters and increase the efficiency of the model. Using multi-layered structures, CNNs can progressively extract complex patterns from low-level data, such as edges and textures, to higher-level features, such as shapes and objects. Pooling layers are also used to reduce computation, prevent overfitting, and shrink the feature graph. ReLU is a widely utilized activation function that incorporates nonlinear elements to improve the network's ability to match complex functions. [6].

2.2.2 Reasons for choosing a particular architecture and its adaptability

For regression and classification problems, the conventional CNN architecture was selected for a number of significant reasons. First, CNNs are capable of extracting spatial characteristics from images with high efficiency. This makes them useful for tasks involving the use of regression models to forecast bounding box coordinates (left, top, right, and bottom) and classification models to distinguish between various cancer cell types. Second, the model's deep structure, which combines many convolution and pooling layers, enhances the model's accuracy and resilience while capturing intricate patterns. In addition, Dropout layers and L2 regularization may help minimize overfitting and guarantee that the model performs well on both training and test sets. You may make better use of a common feature extraction layer and boost productivity for various tasks by creating two distinct models—one for classification and one for regression. This traditional CNN architecture's adaptability enables it to successfully complete people's regression and classification tasks by modifying hyperparameters to fit a range of task criteria [7].

2.3. Model Implementation and Training

2.3.1 Model architecture design

In the model, a combination of multiple convolution layers, pooling layers, and fully connected layers is used to achieve image feature extraction and final prediction. The configuration of the convolutional layers includes several 3x3 convolutional filters. 32 filters are used in each of the first two levels, 64 filters are used in the next two layers, and 128 filters are used in the last two layers. The ReLU activation function is used to establish all convolutional layers, L2 regularization is set to 0.01 and weights are initialized using the "he_uniform" method with padding set to "same" to maintain the input's spatial dimensions.

For every two convolutional layers, a MaxPooling2D layer using a 2x2 pooling window size streamlines the model. The most important features are retained while the computational effort is reduced and the size of the feature maps is reduced thanks to this pooling process. The architecture consists of two layers with 32 and 256 neurons each in the completely connected layers. Eight neurons are used in the output layer for the classification task, which uses the softmax activation function to create a probability distribution for multi-class classification. Four neurons are used in the output layer for the regression task. Selecting the right ReLU activation function is essential since it improves training speed and efficiently addresses the problem of vanishing gradients.

The number of convolutional nuclei in the convolutional layer, the number of neurons in the fully connected layer, and the input and output dimensions of each layer are the primary determinants of the number of parameters in the model. The model has a lot of parameters since it uses fully linked and numerous convolution layers, but the regularization technique efficiently reduces overfitting. The multi-layer nature of the model, particularly the several convolutional layers, contributes to its high computational complexity. Nonetheless, the pooling layer might lessen the feature map's size, which will lessen the computational load on later layers. In addition, selecting the right learning rate and batch size may aid in striking a balance between training's convergence speed and computing complexity.

2.3.2 Training process and hyperparameter adjustment

The regression model uses Mean Squared Error (MSE) as a loss function for training. Predicting the bounding box coordinates (left, top, right, and bottom) is very helpful, and it may also be used to evaluate the discrepancy between the expected and actual values. By averaging the squared difference between the predicted value and the true value, the mean square error may be made more sensitive to large errors. By doing this, the model's training mistakes may be decreased. Applied to classification models, categorical cross-entropy is a helpful tool for multi-class classification issues since it can precisely measure the difference between the correct label and the probability distribution of the model output. The cross-entropy loss efficiently captures the uncertainty in the classification and guides the model's learning process since the output layer of the classification model employs the softmax activation function. Loss function and accuracy are shown in Fig. 1.

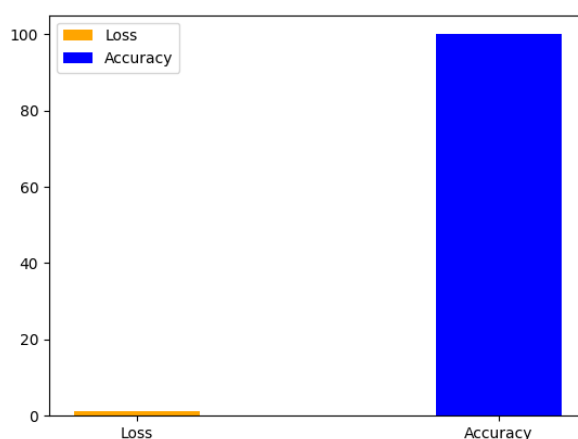


Figure 1. Visualization of loss function and accuracy (Photo/Picture credit: Original)

2.3.3 Optimization algorithm selection

The Adam optimizer is used to choose the optimization algorithm. Adam automatically modifies the learning rate during training, using the benefits of both adaptive learning rate and momentum to accelerate convergence and enhance model performance. Adam works better with sparse gradients and non-stationary targets than the conventional stochastic gradient descent (SGD), particularly in deep learning problems.

2.3.4 Setting of learning rate, batch size, and number of iterations

The learning rate is set to $1e-3$, which works well on many deep-learning tasks and allows for quick updates during the first phase while preserving model stability. The batch size and number of iterations are also taken into consideration. A reasonable batch size finds a decent compromise between memory use and model performance. The batch size is set to 64, meaning that 64 samples are utilized for training each time the model is updated. In the development and debugging stages, it is possible to rapidly confirm the correctness of the model architecture and parameter settings by setting the number of iterations (epochs) to 1, even though this is uncommon in real training. To make sure the model is thoroughly trained, epochs may be progressively raised in successive training based on the model's performance. Fig. 2 displays the loss function and accuracy.

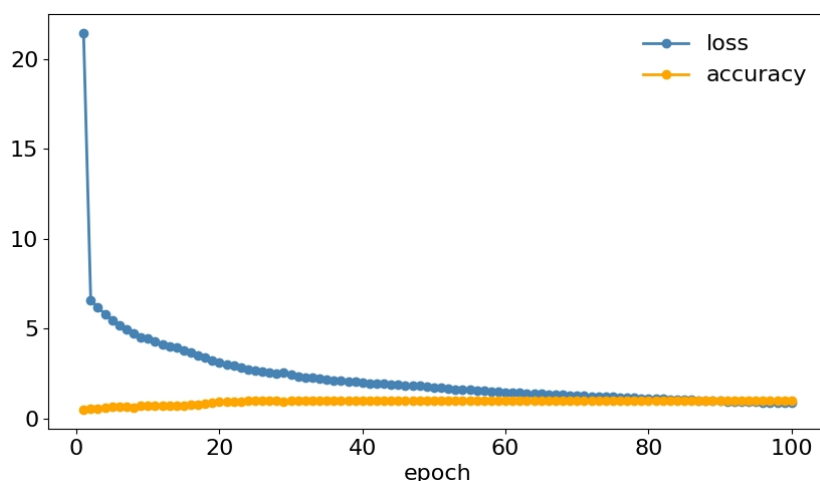


Figure 2. Loss function and accuracy visualization (Photo/Picture credit: Original)

2.4. Classification evaluation index

Among the classification assessment indicators used in the model evaluation process to examine model performance thoroughly are accuracy, accuracy, recall, and F1-score. Accuracy, or the proportion of samples the model correctly predicted out of the total sample, is a useful performance statistic when the class distribution is somewhat stable. The accuracy rate indicates the proportion of correctly predicted positive samples among all samples that the model correctly classified as positive. This metric, which indicates how effectively the model forecasts positive cases, is particularly useful when lowering false positives is a primary priority. On the other hand, the recall rate pertains to the percentage of true positive samples that the model correctly identifies. Because it highlights the model's ability to collect good instances, it is ideal in scenarios where reducing false negatives is crucial. The F1-score combines memory and accuracy into one measure by calculating their harmonic mean. This score is highly useful for managing class imbalances since it provides a more thorough picture of the model's performance than accuracy alone. When combined, these three metrics provide a comprehensive view of the model's performance across many domains [8].

3. Results and Discussions

3.1. Description of Experimental Environment

To ensure efficient model testing and training, this experiment made use of high-performance hardware and software setups. The training process of the deep learning model was significantly accelerated by utilizing a personal computer equipped with an NVIDIA GPU (graphics processing unit) from a hardware standpoint. Python was used as the primary programming language for the software configuration, and the experiment was carried out on a 64-bit Ubuntu operating system. TensorFlow and Keras provided us with the necessary flexibility and capability to create and train deep learning models with reliable results, in particular. TensorFlow and Keras were preferred options

for designing and training deep learning models with dependable outcomes, which provided them with the necessary adaptability and capability. The experiment operated with optimal efficiency due to the combination of hardware and software components. Scikit-learn, Pandas, and NumPy are tools needed for data processing and analysis [9].

3.2. Experimental Control Variables and Settings

The performance of different models and parameter Settings is compared in the experiment through variable control. The experiment conceived multiple model architectures and performed comparative experiments on distinct hyperparameter settings such as the number of convolutional layers, the number of neurons per layer, regularization methods, etc. Different data preprocessing methods, such as normalization, PCA, and t-SNE dimensionality reduction techniques, were compared to observe their effects on model training speed and classification performance. Through these comparative experiments, the optimal model configuration for this task can be identified, providing a basis for subsequent application and optimization [10]. PCA result is shown in Fig. 3. T-SNE visualization is shown in Fig. 4.

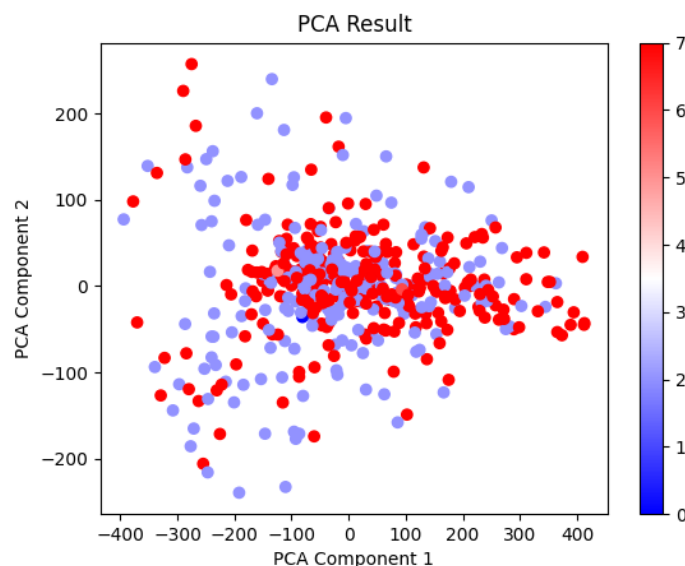


Figure 3. PCA analysis of cancer cell features with color-coded classes (Photo/Picture credit: Original)

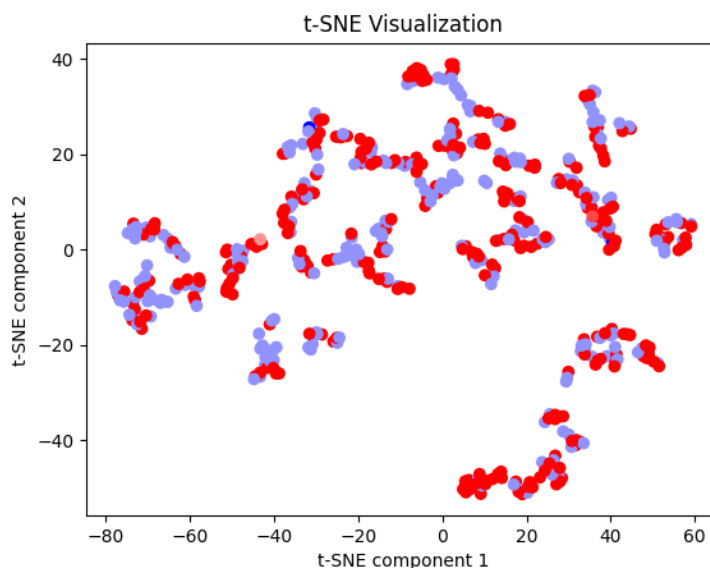


Figure 4. t-SNE visualization of cancer cell features with clustered groups. (Photo/Picture credit: Original)

3.3. Result Analysis

The loss function, or blue curve, rapidly decreases during the first phases of training, showing that the model is picking up new information from the input fast. The loss progressively drops and then stabilizes as the number of training epochs rises, a sign that the model has successfully reduced error. The accuracy (orange curve), on the other hand, begins at a lower level and leveled out ultimately as training goes on, suggesting that the model has hit its peak performance on the training set. CancerEpithelium is shown in Fig. 5. Confusion matrix visualization is shown in Fig. 6.

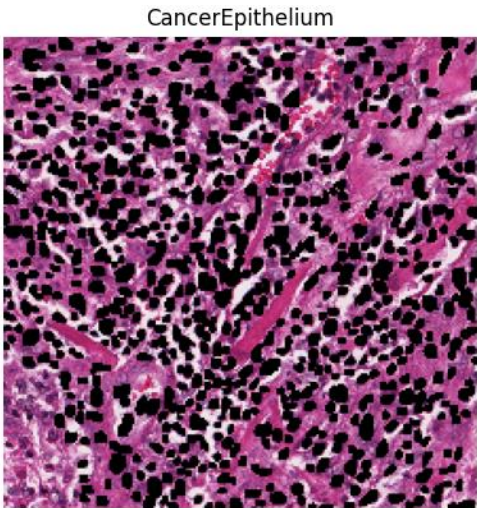


Figure 5. Detected cancer epithelium regions in histologic slide (Photo/Picture credit: Original)

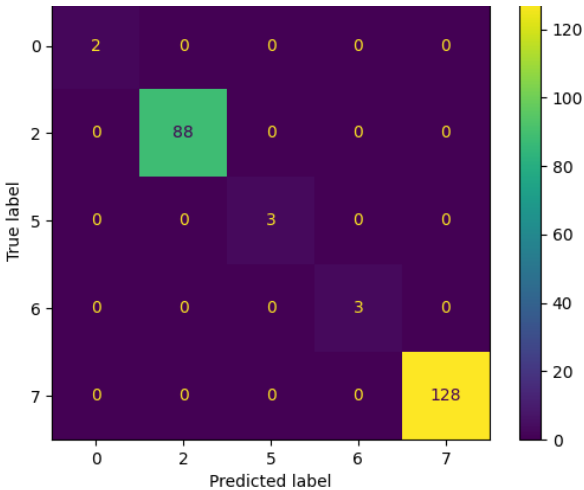


Figure 6. Confusion matrix visualization. (Photo/Picture credit: Original)

It is not feasible to directly compare the performance of the training set with the validation set since the chart only shows the trend of the loss function and accuracy for the training set. Overfitting may be indicated, for example, if the validation set's accuracy is much lower than the training set's, or if the validation set's loss function is larger than the training set's. It is advised to also monitor the model's performance on the validation set to provide a more thorough evaluation of its capacity for generalization.

The model is successfully trained when the loss function continuously decreases, and the accuracy steadily increases during the training period. If the loss keeps going down but the accuracy plateaus, the model could be beginning to overfit due to its recollection of the training set. Underfitting may be indicated if the model shows high loss and poor accuracy in the early phases of training and does not show a discernible improvement as training goes on. Based on the present chart, there is no evidence of underfitting in the model; nevertheless, to determine the danger of overfitting, performance on the validation set must be assessed. Keeping an eye on the validation metrics will

provide information about the model's capacity for generalization and assist in determining if changes are required to increase its robustness.

3.4. Practical Application Scenario

The model has a lot of potential for being applied in cancer screening in real-world settings. The use of CNN models in image recognition and classification tasks has been made possible with the advancement of deep learning, which has demonstrated remarkable performance, making them a solid foundation for medical imaging analysis. The efficiency and accuracy of cancer screening can be enhanced by doctors using this model to identify cancerous lesions earlier and more accurately.

4. Conclusion

The research presented in this paper offers significant advancements in cancer detection through an automated CNN approach, demonstrating its efficacy via extensive testing. By leveraging sophisticated deep learning algorithms and complex image processing techniques, the approach facilitates the comparison and optimization of multiple CNN designs, yielding notable performance improvements. The introduction of batch normalization layers and residual connections has proven to significantly enhance classification accuracy and speed of convergence. Additionally, data augmentation methods such as rotation and random cropping have further improved the model's generalization capabilities. Test results illustrate the effectiveness of the proposed approach in real-time boundary identification and accurate classification of various cancer cell types.

The importance of this research lies not only in its immediate contributions to cancer detection but also in its potential to advance automated diagnostic technologies in medical settings. The exploration of intricate network topologies, which may include advanced convolutional layers and attention mechanisms, aims to bolster the model's ability to detect complex and challenging content. Moreover, extending this approach to other medical imaging modalities, such as CT and MRI scans, represents a crucial step in evaluating its efficacy across diverse medical applications.

Future research will focus on addressing the challenges encountered in real-world scenarios, such as noise handling and data imbalance. Implementing transfer learning in conjunction with advanced data preparation methods and resampling techniques will be key strategies for enhancing model performance, particularly in contexts with limited sample sizes. This comprehensive approach aims to pave the way for more robust and effective automated diagnostic solutions in the field of medical imaging.

References

- [1] Alshehri Alia, and AlSaeed Duaa. Breast Cancer Detection in Thermography Using Convolutional Neural Networks (CNNs) with Deep Attention Mechanisms. *Applied Sciences* 2022, (24), 12922-12922.
- [2] Gonçalves, C. B., Souza, J. R., and Fernandes, H. CNN architecture optimization using bio-inspired algorithms for breast cancer detection in infrared images. *Computers in Biology and Medicine*, 2022, 105, 105205.
- [3] Liang, Z., Nandal, A., Ganchev, T., and Dhaka, A. Breast cancer detection by fusion of deep features with CNN extracted features. *International Journal of Intelligent Systems Technologies and Applications*, 2022, 6, 510-523.
- [4] Hao, R., Khashayar, N., Liu, L., Masoom, H. A., and Khalvati, F. A comprehensive study of data augmentation strategies for prostate cancer detection in diffusion-weighted MRI using convolutional neural networks. *Journal of Digital Imaging*, 2021, 34 (4), 1-15.
- [5] Badawy, M., Balaha, H. M., Maklad, A. S., Almars, A. M., and Elhosseini, M. A. Revolutionizing oral cancer detection: An approach using Aquila and Gorilla algorithms optimized transfer learning-based CNNs. *Biomimetics*, 2023, 6, 499.

- [6] Fan, D., Liang, H., Qu, C., Ma, J., and Hasani, R. A novel approach for early gastric cancer detection using a hybrid of AlexNet, extreme learning machine, and adjusted gorilla troops optimization. *Biomedical Signal Processing and Control*, 2024, 106, 126.
- [7] Liu, Y., Chen, P., Chen, J., Zhang, Y., and Wang, G. Deep learning-based automatic computer-aided diagnosis system for histopathological images of lung cancer. *Journal of Biomedical Informatics*, 2020, 104, 103406.
- [8] Sirinukunwattana, K., Pluim, J. P. W., Chen, H., Qi, X., Heng, P. A., Guo, Y., and Rajpoot, N. M. Gland segmentation in colon histology images: The glas challenge contest. *Medical Image Analysis*, 2017, 35, 489-502.
- [9] Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., and Sánchez, C. I. A survey on deep learning in medical image analysis. *Medical Image Analysis*, 2017, 42, 60-88.
- [10] Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., and Thrun, S. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 2017, 542 (7639), 115-118.