Al-Rafidain Journal of Computer Sciences and Mathematics (RJCM)

www.csmj.mosuljournals.com

Performance Evaluation of Artificial Neural Network Methods Based on Block Machine Learning Classification

Raya Akram Hamdy[1],*Mohammed Chachan Younis[2]

[1, 2] University of Mosul - College of Computer Sciences and Mathematics, Mosul, Iraq
**Corresponding author. Email: mohammed.c.y@uomosul.edu.iq[2]*

Article information

Article history:
Received :31/7/2023
Accepted :13/9/2023
Available online:

Abstract

Abstract— Traditional (pixel-by-pixel) classification techniques are time-consuming, whereas semantic segmentation in machine learning requires assigning class labels to each pixel in an image. This study proposes a block-by-block (5x5 chunks) segmentation method for semantic segmentation, which involves image dissection, feature extraction, and model training based on specific color and textural properties. Thirty cat photos from the Oxford-IIIT Pet dataset were used for evaluation. Five different Artificial Neural Network (ANN) models, including LM, BGFGS, RP, SCG, and GDX, were trained and assessed for both pixel-based and block-based methods. The accuracy of the block-based classification ranges from 82.94% to 85.83%, surpassing the pixel-based approach, which ranges from 70.82% to 76.47%. The processing time for the models also improved with the block-based method. For the pixel-based approach, RP model takes the longest processing time i.e., 242.39 seconds, while GDX model takes the shortest processing time i.e., 49.89 seconds. For the block-based approach, LM model takes the longest processing time i.e., 13.86 seconds, while GDX still has the shortest processing time i.e., 5.98 seconds. Therefore, block-based methods can be seen as more efficient and accurate for classification models. The LM model achieved the highest accuracy on test images, ranging from 94.72% to 89.81%, while the GDX model had the lowest accuracy, ranging from 92.96% to 81.15%. The remaining models, RP, SCG, and BFG, have intermediate levels of accuracy.

Keywords:
ANN Classification, Semantic Segmentation, Features Extraction.

Correspondence:
Author: Mohammed Chachan Younis
Email: mohammed.c.y@uomosul.edu.iq

I. INTRODUCTION

Pixel-based categorization is widely used in digital imaging applications, such as land cover mapping and vegetation studies [1]. By assigning pixel-level labels to objects and regions within an image, these techniques enable machines to comprehend the complex visual world, making them invaluable in areas such as autonomous driving, medical image analysis, and object recognition. However, they face several formidable challenges, including the need for large and diverse annotated datasets, high computational demands, and real-time processing constraints. Additionally, handling occlusions, fine-grained object distinctions, and coping with variations in lighting and viewpoint remain persistent challenges. Addressing these hurdles is essential for the continued advancement and widespread adoption of semantic segmentation methods in various domains. Artificial Neural Networks (ANNs) have proven effective for pixel-based classification, including segmenting animals like cats from their backgrounds [2], [3]. Advances in Machine Learning and deep learning techniques have enabled significant progress in image labeling, object recognition, and scene categorization [4-6]. However, accurately segmenting animals remains challenging due to variations in breeds and sizes. This study evaluates the performance of different ANN training functions for cat object segmentation, emphasizing the importance of selecting the right training function for optimal results. Despite existing challenges [7-9], ongoing research aims to develop more accurate and efficient segmentation methods using machine learning and deep learning [10]. Traditional pixel-based classification methods relying on complex manual features struggle with diverse image characteristics [11], [12]. ANNs, particularly DNNs, CNNs, and RNNs, have shown promise in enhancing image segmentation precision and robustness [1], [13]. However, challenges like hyperparameter selection, overfitting, and underfitting need further attention [14]. ANNs for pixel-based classification and image segmentation are active research areas with transformative potential across domains [15], [16]. The proposed study evaluates various ANN training functions for cat object segmentation, introducing a block-based approach to address the time-consuming nature of pixel-by-pixel classification while maintaining accuracy. Comparative analysis between pixel-by-pixel processing and the block-based method demonstrates the superiority of the block-by-block strategy. The study investigates different ANN training functions and feature extraction techniques, contributing to improved cat object segmentation. The objectives of this proposed study is to find a better way to classify images using artificial neural networks to surpass the accuracy and efficiency of the prior studies. Following are the objectives of the study:

1. To propose state of the art architecture for the image classification model that incorporates block-based segmentation techniques to reduce processing time and improve efficiency.
2. To input and pre-process the data i.e., loading input images, dividing image into fixed size blocks, inputting the blocks of pixels into the proposed architecture.
3. To extract different features such as contrast, correlation, energy, homogeneity, hue variation, saturation variation, and value variation of HSV images etc. for each block within the block-based segmentation framework. These features will be used as input for the classification.
4. To train artificial neural network classification models using different sets of layers i.e., input, hidden and output layers.
5. To compare the performance of various backpropagation algorithms such as Levenberg-Marquardt, BFGS Quasi-Newton, Resilient Backpropagation, Scaled Conjugate Gradient, and Variable Learning Rate Backpropagation.
6. To visualize the results and analyze the performance of the training model using classification accuracy of trained models, ROC (Receiver operating characteristic) curves, and confusion matrices.
7. To evaluate the trained models performance for test data through classification accuracy metrics, predicted images and confusion matrices, for predicting three test images considering the block-based segmentation technique.

The proposed study aims to evaluate the effectiveness of various ANN training functions for separating cat objects from backgrounds. Five training functions will be compared using a dataset of cat images. MatLab software, along with Image Processing and Neural Network Toolboxes, will be used for preprocessing, feature extraction, model training, and testing [17], [18]. Evaluation criteria such as precision, recall, and F1 scores will be employed to assess the performance of the ANN classification models [19]. The study has limitations including the dependence on image quality and resolution for block-based segmentation accuracy, potential impact of dataset size and diversity on model performance measures, focus on cat objects rather than other animal species, and potential constraints on resources for conducting experiments.

The related work in this study focuses on semantic segmentation, its significance, characteristics, challenges, and applications. Semantic segmentation involves classifying individual pixels in an image into distinct categories [20-24]. Advancements in artificial neural networks, particularly convolutional neural networks (CNNs), have improved the performance of semantic segmentation tasks [25]. Several studies and benchmarks have addressed the complexities of region segmentation and class imbalance [26], [27]. Accurate segmentation requires considering the image context, selecting robust features, and balancing speed and precision [24], [28-30]. Semantic segmentation has diverse applications, including object removal, scene interpretation, medical image analysis, and object detection [31]. Combining semantic segmentation with neural network classification and multi-modal data fusion shows promise for improving segmentation performance [32]. Artificial neural networks (ANNs) have been widely used in previous studies for tasks like object detection and image classification [2], [33]. Various factors affect the performance of an ANN model, such as training data, network architecture, activation functions, and hyperparameters. Evaluation of ANN models for pixel-based categorization involves metrics like accuracy, precision, recall, and F1 score, as well as class-specific metrics and confusion matrices [31]. Different ANN models have been explored in various classification tasks, including data categorization, phishing detection, rig state classification, thermal image flaw detection, music classification, plantar pressure image categorization, and medical image classification [31], [34-38]. Fully connected neural networks, convolutional neural networks (CNNs), and conditional generative adversarial networks (cGANs) are commonly used ANN models [39-41]. CNNs, with their efficient feature extraction capabilities, are particularly popular for image classification and object detection. Semantic segmentation, which involves classifying individual pixels into distinct object classes, is an important component of computer vision tasks, and ANNs have shown promising results in this area [41].

The proposed study consists of well-organized research presented in four sections. Section 1 introduces the purpose and objectives of the research and also provides the theoretical background and literature review. Section 2 explains the methodology used for data collection and analysis. Section 3 presents the research findings, and discusses and interprets the results. Section 4 summarizes the main conclusions and suggests areas for future research. The study follows a logical sequence and concludes with a list of references.

II. METHODS

The study aims to compare the performance of ANN models for pixel-based and block-based classification tasks using the methodology shown in Fig. 1. A dataset with original images and ground truth labels was collected and split into training and validation sets as shown in Fig. 1. The trained models were evaluated using a separate test dataset. Obtaining ground truth images with pixel labels can be time-consuming, so the dataset was carefully prepared to include original images and corresponding ground truths. The models were trained and fine-tuned using the training and validation sets [42], [43], and then tested on unknown images using a separate test dataset. The high-level structure of our proposed model is shown in Fig. 1, which highlights the various stages involved in the training, validation, and testing of the model. By following this approach, we aimed to develop high-performance semantic segmentation models that can accurately identify and categorize multiple objects in an image, even if they span multiple regions at the pixel level. Fig. 1 illustrates the proposed methodology for transforming images used for training and validation into 5x5 pixel blocks and extracting 16 different texture and color-based features from each block segment. The extracted features are compiled as a feature vector along with the corresponding class label for each block. Five different training functions are used to train different ANN models using features extracted from the labeled image blocks. The input layer of the ANN architecture receives the features of the pixel blocks, and the hidden layer of the ANN architecture computes and processes the features to deliver the output at the output layer of the ANN architecture. In the next phase, we applied unknown test images to the previously trained ANN classifiers i.e., LM, RP, SCG, BFG, and GDX. We evaluated these classifiers and analyzed the results obtained from them to measure their performance in pixel-based classification.

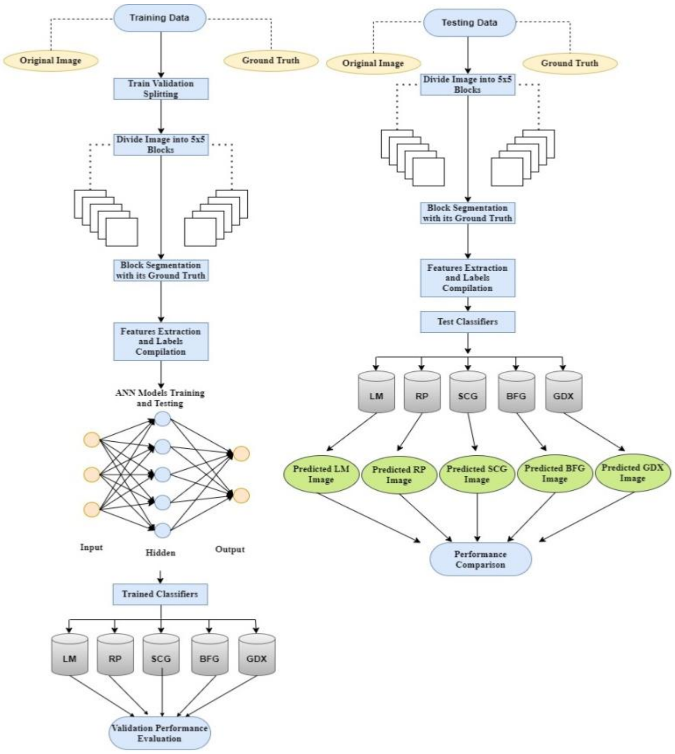
Training Data
Original Image
Ground Truth
Train Validation Splitting
Divide Image into 5x5 Blocks
Block Segmentation with its Ground Truth
Features Extraction and Labels Compilation
ANN Models Training and Testing
Input
Hidden
Output
Trained Classifiers
LM
RP
SCG
BFG
GDX
Validation Performance Evaluation
Testing Data
Original Image
Ground Truth
Divide Image into 5x5 Blocks
Block Segmentation with its Ground Truth
Features Extraction and Labels Compilation
Test Classifiers
LM
RP
SCG
BFG
GDX
Predicted LM Image
Predicted RP Image
Predicted SCG Image
Predicted BFG Image
Predicted GDX Image
Performance Comparison

FIG. 1. TOP-LEVEL ARCHITECTURE OF THE SYSTEM

Data Visualization

The proposed study utilized the Oxford-IIIT Pet dataset for cat segmentation [44]. Thirty cat images were selected and divided into 5x5 blocks. Despite the dataset's large size (around 800 MB), the project's objective was to perform cat segmentation and isolate the cat from the background. From each block, 16 features were extracted to create a Feature Vector using MatLab. ANN models were trained using this Feature Vector to isolate the cat from the background. The dataset's original purpose was breed identification, but the study focused on cat segmentation. Fig. 2 shows the images (left) and corresponding ground truth data (right).

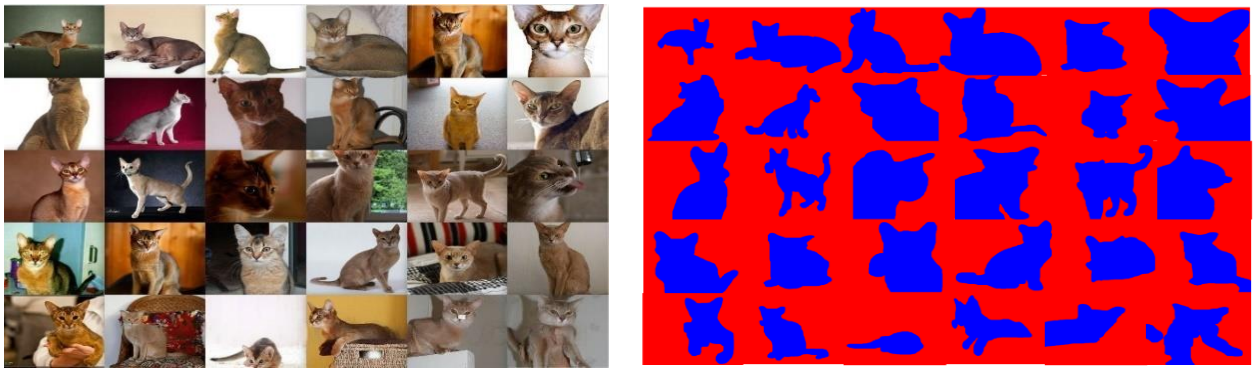

FIG. 2. CAT IMAGES SAMPLES AND GROUND TRUTH IMAGES FROM THE DATASET

The dataset used in this study [44] contained original images and corresponding ground truth images for cat object classification. A subset of 30 cat images was selected, and the ground truth images had three classes i.e., cat, border, and background. The cat boundary class was merged with the cat object class to simplify the dataset into two classes. The simplified dataset was used for training and testing the ANN models in cat image segmentation. Fig. 3 illustrates the original and merged ground truth images. Fig. 3 (left) shows the original ground truth image with the three classes, while Fig. 3 (middle) displays the image after merging the cat boundary and cat object classes, resulting in two consolidated classes. The dataset of 30 cat images was selected and colored ground truth images were generated for the proposed study. The cat object and its boundary classes were merged into one class, while the background was marked separately. The resulting ground truth images were displayed with the cat object in blue and the background in red, as shown in Fig. 3 (right). MatLab software was utilized for image processing.

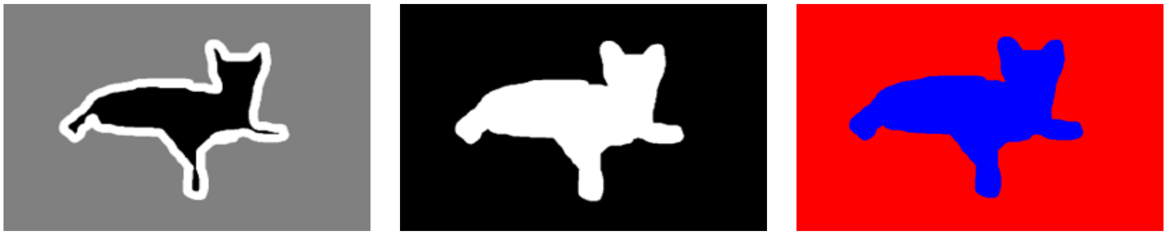

FIG. 3. (LEFT) CAT IMAGE WITH 3 CLASSES, (MIDDLE) CAT IMAGE WITH 2 CLASSES, (RIGHT) GROUND TRUTH IMAGE.

Feature Extraction from the Preprocessed Image

The training images were divided into 5x5 blocks, and 16 color and textural features were extracted from each block. These features included statistical measures such as mean and standard deviation for RGB channels, as well as hue, saturation, brightness, contrast, correlation, energy, and homogeneity. The extracted features were used to create feature vectors for each block, along with class labels indicating cat objects or backgrounds. These feature vectors and labels were utilized to train the Artificial Neural Network (ANN) models. This training process enabled the models to learn the relationships between features and class labels, enabling accurate predictions during segmentation. The following are 16 image processing attributes that are used in this study to acquire the important distinctive features of the image and their detailed descriptions are beyond the scope of this study: Contrast, Correlation, Energy, Homogeneity, Hue Variation, Saturation Variation, Intensity Value Variation, Hue Standard Deviation, Saturation Standard Deviation, Value (Intensity) Standard Deviation, Hue Mean, Saturation Mean, Value (Intensity) Mean, Hue Skewness, Saturation Skewness, and Value (Intensity) Skewness.

ANN Models

Artificial Neural Network (ANN) classification is a machine learning technique that categorizes data into predefined classes. ANNs consist of interconnected neurons organized into layers, processing input data through weighted sums and activation functions. During training, the network's weights and biases are adjusted using optimization algorithms like stochastic gradient descent [45]. Trained ANNs can classify new data by inputting it into the network. While ANNs are powerful for complex datasets [46], they require sufficient training data and can be computationally intensive. Interpreting their inner workings can also be challenging. ANN classification is widely used in image classification applications [7].

Levenberg-Marquardt Back Propagation (LM-Model)

The Levenberg-Marquardt (LM) algorithm is a widely used training function for Artificial Neural Networks (ANNs) that improves convergence speed and stability by combining gradient descent with a trust-region approach [47]. It effectively handles highly nonlinear relationships between input and output variables by adapting the step size based on the error surface curvature. The LM algorithm finds applications in regression and classification tasks, including image recognition, financial forecasting, and medical diagnosis [48]. It is integrated into popular ANN software packages such as MatLab's Neural Network Toolbox and the open-source library Keras [49]. The algorithm aims to minimize the cost function J(w), which measures the discrepancy between predicted output y and actual output t for a given input x, by adjusting the ANN weights w give in equation (1).

$$w(k+1) = w(k) - [J'(w(k))' J'(w(k)) + \mu I]\hat{}(-1) J'(w(k))' \quad ...(1)$$

In the LM algorithm's equation (1) represents the weight vector at iteration k as w(k), and the damping parameter balancing gradient descent and trust-region approaches is represented as μ. The Jacobian matrix of the cost function with respect to the weights is denoted as J'(w), and I represents the identity matrix. The LM algorithm typically achieves superior performance compared to standard backpropagation algorithms like gradient descent and conjugate gradient when dealing with highly nonlinear relationships between input and output variables [50]. However, it should be noted that the LM algorithm may require additional computational resources due to the computation and inversion of the Jacobian matrix.

Broyden Fletcher Goldfarb Shanno (BGFGS- Model)

The BFGS (Broyden-Fletcher-Goldfarb-Shanno) algorithm is a widely used optimization method for training artificial neural networks [51]. It iteratively updates the estimate of the inverse Hessian matrix by utilizing gradient information, improving the convergence to the optimal solution [52]. The BFGS algorithm is an improvement over the original quasi-Newton method, incorporating a rank-two correction formula to enhance the Hessian approximation. The update rule for the BFGS algorithm is given by equation (2).

$$H\_k + 1 = H\_k + [(y\_k * y\_k') / (y\_k' * s\_k)] - [(H\_k * s\_k * s\_k' * H\_k) / (s\_k' * H\_k * s\_k)] \quad ...(2)$$

In equation (2), H_k represents the inverse Hessian approximation at iteration k, y_k represents the difference between the gradients at iterations k+1 and k, s_k represents the difference between the parameters at iterations k+1 and k, and g(x_k) is the gradient of the objective function at iteration k. The BFGS algorithm is effective for training artificial neural networks [53], especially for small to medium-sized networks, and it has faster convergence compared to other optimization methods like conjugate gradient descent and Levenberg-Marquardt. In MatLab, the BFGS algorithm is implemented as the 'trainbfg' function in the Neural Network Toolbox.

Resilient Back Propagation (RP-Model)

Resilient Backpropagation (RPROP) is a popular training algorithm for feed forward Artificial Neural Networks (ANN) that is efficient, robust, and parameter-free. It adjusts the weight update step size using only the signs of the gradients, making it suitable for sparse, noisy, and large-scale optimization problems. Proposed by [54], RPROP has been widely used in pattern recognition, image processing, and natural language processing. Equation (3) defines the weight update rule used by the algorithm.

$$\Delta w\_ij(n) = -\eta(n) * sign(\partial E / \partial w\_ij) \quad ...(3)$$

Equation (3) defines the weight update (Δw_ij(n)) between neurons i and j in iteration n. RPROP adjusts the step size (η(n)) based on gradient signs: increasing by a small factor if the sign remains the same, and decreasing by a larger factor if the sign changes. This adaptive approach ensures fast convergence and avoids local minima. RPROP is a robust and efficient algorithm that does not require manual parameter tuning, making it suitable for various problems [55]. However, for highly nonlinear and complex problems, alternatives like the Levenberg-Marquardt algorithm may be more suitable, as RPROP may not always reach the global optimum.

Scaled Conjugate Gradient Back Propagation (SCG-Model)

The Scaled Conjugate Gradient (SCG) algorithm, introduced by [56], is a fast and efficient optimization method commonly used in neural network training. It is particularly effective for networks with numerous parameters, as it achieves rapid convergence while minimizing memory and computation requirements. SCG involves scaling the gradient using a diagonal matrix D, which is updated iteratively. The weight update is obtained by minimizing a quadratic approximation of the error function using the scaled gradient and search direction. Equation (4) provides the weight update equation for SCG.

$$\Delta w \ = \ -\eta g \ / \ (D \ + \ \lambda I) * g\_prev \qquad ...(4)$$

In equation (4), the Δw is the weight update vector, η is the learning rate, g is the gradient vector, D is the diagonal matrix of scaling factors, λ is a damping factor, I is the identity matrix, and g_prev is the previous gradient vector. The SCG algorithm has been implemented in several software packages for training neural networks, including the Neural Network Toolbox in MatLab and the open-source library PyBrain in Python.

Variable Learning Rate Back Propagation (GDX-Model)

The traingdx function is a popular training algorithm in Artificial Neural Networks (ANNs) for backpropagation-based learning, available in the MatLab neural network toolbox [57]. It employs the gradient descent with momentum algorithm, based on the Levenberg-Marquardt method, leading to faster convergence compared to other gradient-based training algorithms. By computing the gradient of the error function with respect to each weight and bias, traingdx updates them in the direction of steepest descent. To avoid local minima, the algorithm includes a momentum term. Additional features like adaptive learning rate adjustment, weight and bias regularization, and automatic input data scaling contribute to enhanced convergence speed and network accuracy. Equation (5) provides the update rule for the weights and biases in traingdx.

$$w(t+1) \ = \ w(t) \ - \ delta\_w(t) \, b(t+1) \ = \ b(t) \ - \ delta\_b(t) \qquad ...(5)$$

In equation (5), w is the weight matrix, b is the bias vector, t is the iteration number, and delta_w and delta_b are the weight and bias update terms calculated using the gradient and momentum information.

III. RESULTS AND DISCUSSION

In the results section, we will analyze and present the outcomes of training and testing various models. We will provide data visualizations, training procedures, ROC curves, confusion matrices, and accuracy scores for the classification of cat and background images using the training dataset. Additionally, we will evaluate the models' performance on unseen data by generating ROC curves, confusion matrices, and accuracy scores using the testing dataset.

Training Model Evaluation

To evaluate model performance and determine the best strategy, this study employed five training algorithms (LM, BFG, SCG, RP, and GDX) implemented using MatLab functions. The models were evaluated on the training dataset using metrics such as accuracy, ROC and Confusion Matrices. Accuracy measures the percentage of correctly classified instances, ROC plot and Confusion matrices show class level results. By comparing the results, the most effective approach can be determined for accurate predictions in future scenarios.

Accuracies of Models on the Training dataset

Table 1 presents the accuracy and processing time of five classification models, namely trainlm, trainrp, trainscg, trainbfg, and traingdx, on the training dataset. Accuracy represents the ratio of correctly classified instances to the total number of instances, while processing time indicates the duration in seconds required by each model to train on the dataset.

TABLE 1. ACCURACIES AND PROCESSING TIME OF CLASSIFICATION MODELS FOR TRAINING DATASET

Classification Model	Accuracy (%)	Time (Sec)
Levenberg-Marquardt Back Propagation (trainlm)	85.83	13.87
Resilient Back Propagation (trainrp)	82.94	8.04
Scaled Conjugate Gradient Back Propagation (trainscg)	83.74	8.12
Broyden Fletcher Goldfarb Shanno (trainbfg)	84.36	8.85
Variable Learning Rate Back Propagation (traingdx)	76.00	5.98

The trainlm algorithm attained the highest accuracy with 85.83%, followed by trainbfg, trainscg, and trainrp with 84.36%, 83.74%, and 82.93% accuracy, respectively. The lowest accuracy obtained by the traingdx algorithm was 76.00%. The longest processing time was 13.87 seconds for trainlm, followed by 8.85 seconds for trainbfg, 8.12 seconds and 8.04 seconds for trainscg and trainrp, respectively, and 5.98 seconds for traingdx. Among the models evaluated on the training dataset, the trainlm algorithm is the most accurate but also the slowest, while the traingdx algorithm is the quickest but least accurate.

Accuracy and Time Comparison of Pixel based Segmentation and Block based Segmentation Approach

The study compares pixel-by-pixel classification and a proposed block-based method for cat image classification. Using the same dataset, the results indicate that block-based segmentation outperforms pixel-by-pixel processing in terms of classification accuracy across all five training functions used for ANN training. Additionally, the processing time for pixel-by-pixel classification is significantly longer compared to block-based segmentation, as shown in Table 2, which compares accuracy and time values for both methods.

TABLE 2. COMPARISON OF ACCURACY AND PROCESSING TIME FOR PIXEL-BASED AND BLOCK-BASED METHODS

Classification Model	Pixel Based Accuracy (%)	Block Based Accuracy (%)	Pixel Based Time (Sec)	Block Based Time (Sec)
trainlm	76.47	85.83	89.96	13.87
trainrp	75.67	82.94	242.39	8.04
trainscg	74.71	83.74	126.59	8.12
trainbfg	75.44	84.36	198.28	8.85
traingdx	70.82	76.00	49.89	5.98

Table 2 compares the accuracy and processing time of pixel-based and block-based methods for classification models. Five models i.e., trainlm, trainrp, trainscg, trainbfg, and traingdx are evaluated in both approaches. Pixel-based accuracy ranges from 70.82% to 76.47%, while block-based accuracy ranges from 82.94% to 85.83%. All models demonstrate improved accuracy in the block-based approach. Processing time is also presented in the table. For the pixel-based approach, trainlm has the longest processing time (89.96 seconds), while traingdx is the fastest (49.89 seconds). In the block-based approach, trainrp has the longest processing time (242.39 seconds), and traingdx remains the quickest (5.98 seconds). Therefore, block-based methods are more efficient and accurate for classification models.

Evaluation of Models for Testing Dataset

The study introduced a block-based segmentation method and trained five models with different training functions. These models are capable of separating the cat object from the background in any unknown cat image. Performance evaluation and comparison were conducted using three sample test images, as shown in Fig. 5 at top position. Ground truth images, displayed in Fig. 4 at the bottom, were used to visually assess the segmentation performance by comparing them with the predicted images generated by the trained models. The accuracy and effectiveness of the segmentation method were evaluated based on this comparison.

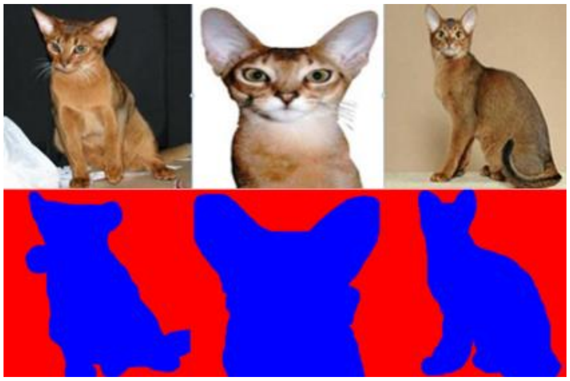

FIG. 4. TESTING IMAGES (TOP) AND GROUND TRUTH IMAGES (BOTTOM)

Table 3 presents the accuracy of different classification models on three test images. Five training functions were used to train the models, and their accuracy was evaluated based on the test images. The table shows the accuracy percentages for each model and test image. The trainlm model achieves the highest accuracy across all three test images, with percentages of 94.71%, 84.92%, and 89.81% for Test Images 1, 2, and 3, respectively. The traingdx model has the lowest accuracy, with percentages of 92.96%, 66.90%, and 81.14% for Test Image 1, Test Image 2, and Test Image 3, respectively. The remaining models (trainrp, trainscg, and trainbfg) demonstrate intermediate levels of accuracy, as shown in Table 3.

TABLE 3. ACCURACY COMPARISON OF MODELS FOR PREDICTION OF TEST IMAGES

Classifier Type	Test Image 1 Accuracy (%)	Test Image 2 Accuracy (%)	Test Image 3 Accuracy (%)
Trainlm	94.72	84.92	89.81
Trainrp	94.63	77.48	86.17
Trainscg	94.30	81.28	87.38
Trainbfg	94.70	84.80	87.79
Traingdx	92.96	66.90	81.14

Accuracy Metrics of LM Classifier to Predict Test Image 1

The ROC curve of the LM classifier shown in Fig. 5, concludes that the classifier performs well for both the background and cat classes. The AUC (Area under curve) is high, indicating high accuracy of the classifier. The background class is predicted correctly 17277 times and misclassified as cat 434 times. The cat class is predicted correctly 13316 times and misclassified as background 1273 times. The overall accuracy of the LM classifier is 94.7%.

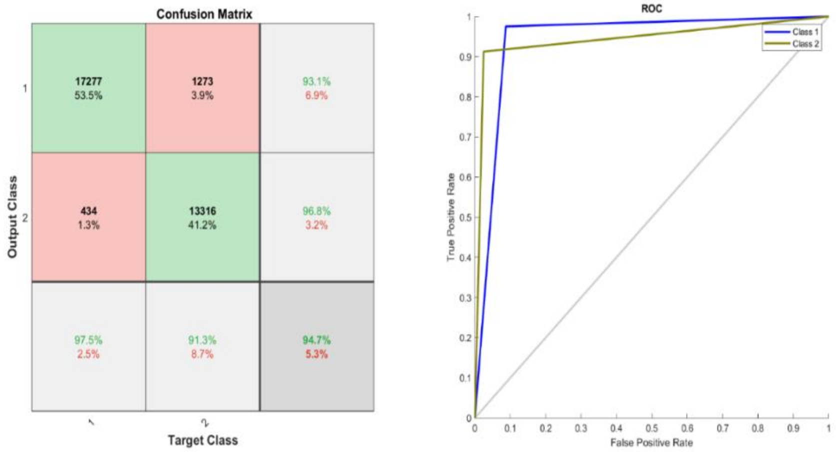

FIG. 5. CONFUSION MATRIX AND ROC CURVE OF LM CLASSIFIER FOR PREDICTION OF IMAGE 1

Accuracy Metrics of LM Classifier to Predict Test Image2

The performance of the LM classifier was evaluated using test image 2 as input. Fig. 6 displays the accuracy metrics, including the confusion matrix and ROC curve. The AUC values for both classes 1 and 2 indicate high accuracy. The roc curve for class 1 background closely resembles the ideal curve, while the roc curve for class 2 cat has a breakpoint at 0.8 TPR. In the LM classifier, class 1 was predicted correctly 41639 times but misclassified as class 2 422 times. Similarly, class 2 was predicted correctly 73003 times but misclassified as class 1 19936 times.

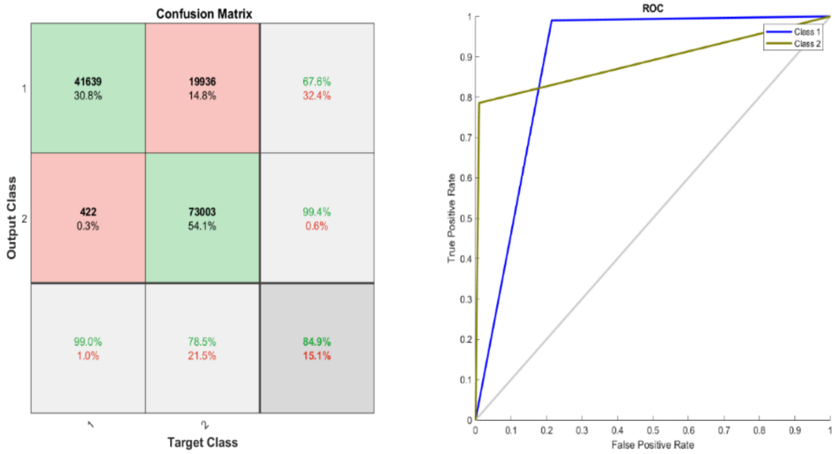

FIG. 6. CONFUSION MATRIX AND ROC CURVE OF LM CLASSIFIER FOR PREDICTION OF IMAGE 2

Accuracy metrics of LM classifier to Predict Test Image 3

Fig. 7 displays the confusion matrix and ROC curve of the LM classifier when it was fed with test image 3 to the ANN classifier. The LM classifier achieves higher levels of accuracy as the Area under the curve increases. The confusion matrix reveals that the background, i.e., class 1, is predicted correctly as class 1, i.e., 97074 times, and as class 2, i.e., cat, 3231 times. The Cat, representing class 2, is incorrectly predicted as background 18701 times and correctly as cat 68994 times. The overall precision of this classifier is 88.3%.

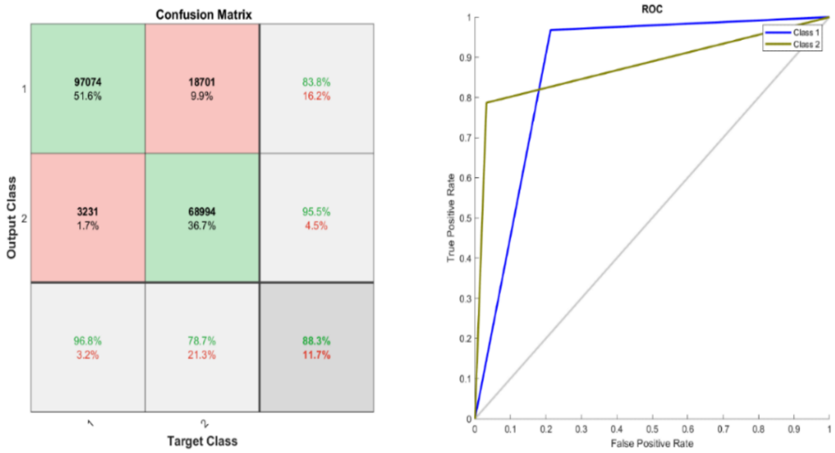

FIG. 7. CONFUSION MATRIX AND ROC CURVE OF LM CLASSIFIER FOR PREDICTION OF IMAGE 3

Visual Comparison of Results of Test Image 1

Fig. 8 presents the predicted images of all the classifiers on Test image 1, where the top left image represents the actual ground truth image, while the top middle, top right, bottom left, bottom middle, and bottom right images represent the predicted images after testing the trained models for BFG, GDX, LM, RP, and SCG, respectively. The predicted images clearly indicate that the LM model provides the most similar predicted image to the actual ground truth image after the classification. This observation is consistent with the classification accuracy values presented in Table 3, where the LM model achieves the highest classification accuracy of 94.72% for Test image 1.

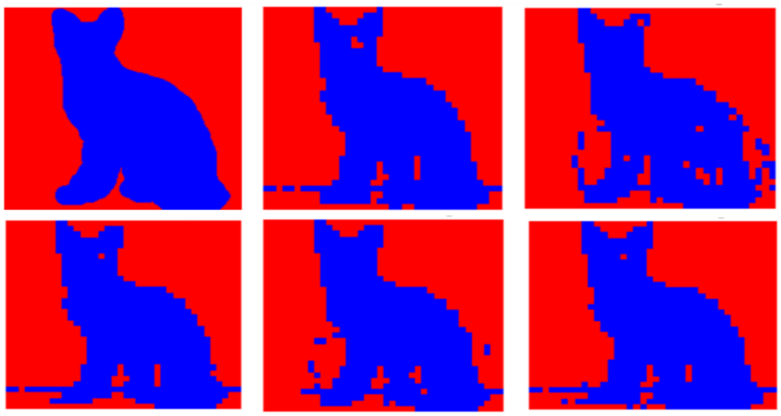

FIG. 8. (TOP LEFT) GT, (TOP MIDDLE) BFG, (TOP RIGHT) GDX, (BOTTOM LEFT) LM, (BOTTOM MIDDLE) RP, (BOTTOM RIGHT) SCG.

Visual Comparison of Results of Test Image 2

Fig. 9 displays the predicted images of all classifiers on Test image 2. The actual ground truth image is shown in the top left, while the top middle, top right, bottom left, bottom middle, and bottom right show the predicted images after testing the trained models for BFG, GDX, LM, RP, and SCG, respectively. It is clear from the predicted images that the LM model provides the most similar predicted image to the actual ground truth image after classification. This is consistent with the classification accuracy values presented in Table 3, where the LM model achieves the highest classification accuracy of 84.92% for Test image 2.

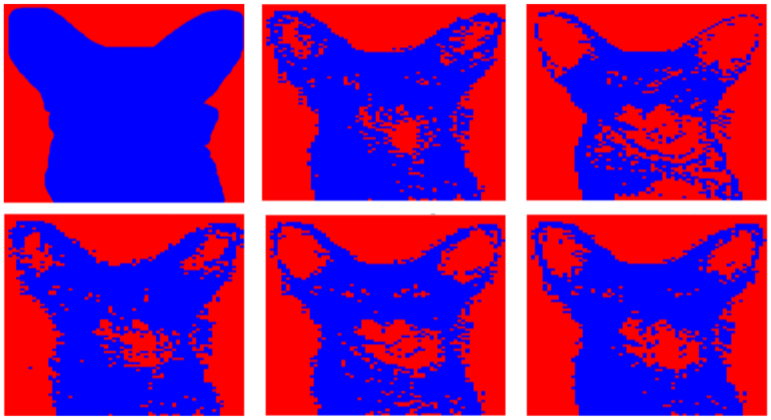

FIG. 9. (TOP LEFT) GT, (TOP MIDDLE) BFG, (TOP RIGHT) GDX, (BOTTOM LEFT) LM, (BOTTOM MIDDLE) RP, (BOTTOM RIGHT) SCG.

Visual Comparison of Results of Test Image 3

In Fig. 10, the predicted images of all classifiers on Test image 3 are presented. The top left image is the actual ground truth image, while the top middle, top right, bottom left, bottom middle, and bottom right images are the predicted images after testing the trained models for BFG, GDX, LM, RP, and SCG, respectively. It is evident from the predicted images that the LM model produces the most similar predicted image to the actual ground truth image after the classification. This also confirms the classification accuracy values presented in Table 3, where the LM model achieves the highest classification accuracy of 89.80% for Test image 3.

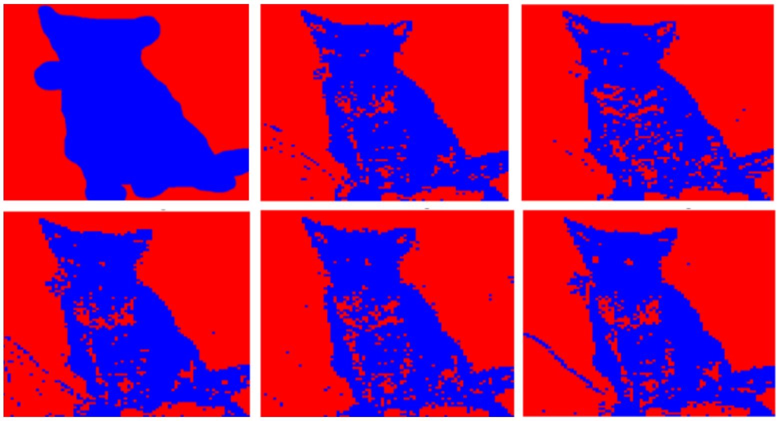

FIG. 10. (TOP LEFT) GT, (TOP MIDDLE) BFG, (TOP RIGHT) GDX, (BOTTOM LEFT) LM, (BOTTOM MIDDLE) RP, (BOTTOM RIGHT) SCG.

Comparison of Classifier Performance for Test Image Prediction

The accuracy comparison of all the tested classifiers for the prediction of test images is presented in Fig. 11. It is evident from the Fig. 11 that the trainlm classifier outperformed all other classifiers with a significant margin for all three test images. On the other hand, traingdx was the least effective classifier among all the tested classifiers.

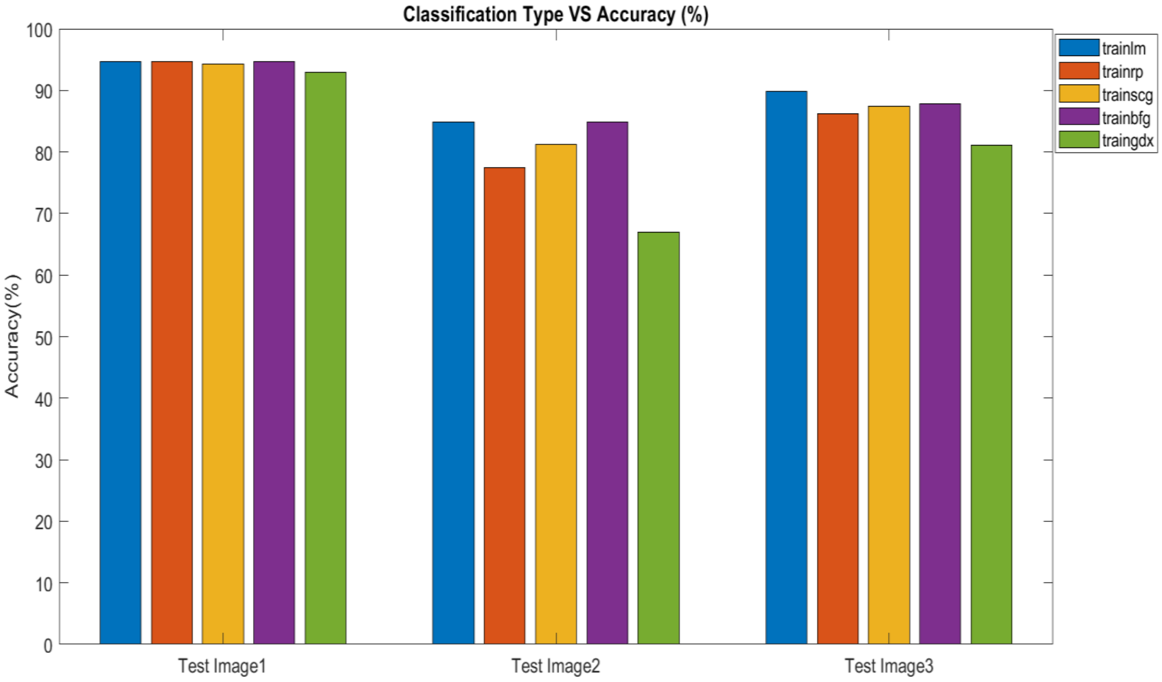

FIG. 11. ACCURACY COMPARISON OF ALL CLASSIFIERS FOR THE THREE TEST IMAGES.

Conclusion

This study is concluded based on the achieved objectives. The results demonstrate that accuracy percentages have improved by transitioning from pixel-based to block-based segmentation. The block-based method also proves to be less time-consuming where traingdx is the fastest (49.89 seconds) in Pixels-based method while traingdx remains the quickest (5.98 seconds) for block-based. The block-based segmentation method shows potential for training accurate classifiers to distinguish cat objects from their image backgrounds. Visual comparisons of test images further support this conclusion, indicating the similarity between predicted and ground truth images after classification. A comparison between pixel-by-pixel and block-based neuromas ANN models reveals that the 5x5 block-based approach is more efficient than the pixel-based one. It is important to note that among the evaluated models, the most accurate algorithm may also be the slowest, while the quickest algorithm may be the least accurate, suggesting the absence of an optimal classifier in terms of both accuracy and time.

The proposed study offers several recommendations for future research. Based on the findings future research can focus on improving the efficiency of image segmentation tasks using ANN models. This can be achieved by exploring hybrid neural network models that combine the strengths of different ANN architectures, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs). Investigating transfer learning techniques for image classification tasks is a prospective area of future research. Transfer learning, which involves pre-training an ANN model on a large dataset and fine-tuning it for a specific task, has shown promise in reducing the amount of training data required and improving the accuracy of ANN models in image segmentation tasks. Developing more efficient and automated hyperparameter optimization techniques for ANN models is worth exploring. Hyperparameter optimization is a crucial step in designing ANN models, but it can be time-consuming and requires manual tuning. Future research can examine the applicability of image segmentation techniques in various fields, such as medical imaging, robotics, and autonomous driving. Precise and efficient object segmentation is essential in these domains for decision-making and navigation purposes.

Acknowledgement

The authors would express their thanks to the College of Computer Sciences and Math. - University of Mosul to support this research.

References

[1] Bello, R.-W., Mohamed, A.S.A., and Talib, A.Z.: ‘Contour extraction of individual cattle from an image using enhanced Mask R-CNN instance segmentation method’, IEEE Access, 2021, 9, pp. 56984-57000. https://doi.org/10.1109/ACCESS.2021.3072636

[2] Argatov, I.: ‘Artificial Neural Networks (ANNs) as a novel modeling technique in tribology’, Frontiers in Mechanical Engineering, 2019, 5, pp. 30. https://doi.org/10.3389/fmech.2019.00030

[3] Fu, H., Song, G., and Wang, Y.: 'Improved YOLOv4 marine target detection combined with CBAM', Symmetry, 2021, 13, (4), pp. 623. https://doi.org/10.3390/sym13040623

[4] Wurm, M.F., and Caramazza, A.: 'Two 'what'pathways for action and object recognition', Trends in cognitive sciences, 2022, 26, (2), pp. 103-116. https://doi.org/10.1016/j.tics.2021.10.003

[5] Wei, X., Phung, S.L., and Bouzerdoum, A.: 'Visual descriptors for scene categorization: experimental evaluation', Artificial Intelligence Review, 2016, 45, pp. 333-368. https://doi.org/10.1007/s10462-015-9448-4

[6] Oh, S.-I., and Kang, H.-B.: 'Object detection and classification by decision-level fusion for intelligent vehicle systems', Sensors, 2017, 17, (1), pp. 207. https://doi.org/10.3390/s17010207

[7] LeCun, Y., Bengio, Y., and Hinton, G.: 'Deep learning', nature, 2015, 521, (7553), pp. 436-444. https://doi.org/10.1038/nature14539

[8] Cui, B., Chen, X., and Lu, Y.: 'Semantic segmentation of remote sensing images using transfer learning and deep convolutional neural network with dense connection', Ieee Access, 2020, 8, pp. 116744-116755. https://doi.org/10.1109/ACCESS.2020.3003914

[9] Cui, X., Lu, C., and Wang, J.: '3D semantic map construction using improved ORB-SLAM2 for mobile robot in edge computing environment', IEEE Access, 2020, 8, pp. 67179-67191. https://doi.org/10.1109/ACCESS.2020.2983488

[10] Jain, A.K.: 'Data clustering: 50 years beyond K-means', Pattern recognition letters, 2010, 31, (8), pp. 651-666. https://doi.org/10.1016/j.patrec.2009.09.011

[11] Yang, A., Yang, X., Wu, W., Liu, H., and Zhuansun, Y.: 'Research on feature extraction of tumor image based on convolutional neural network', IEEE access, 2019, 7, pp. 24204-24213. https://doi.org/10.1109/ACCESS.2019.2897131

[12] Fei-Fei, F.: 'Perona, 2006 Fei-Fei L., Fergus R., Perona P', One-shot learning of object categories, IEEE Trans. Pattern Anal. Mach. Intell, 2006, 28, (4), pp. 594-611. https://doi.org/10.1109/TPAMI.2006.79

[13] Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., and Pietikäinen, M.: 'Deep learning for generic object detection: A survey. arXiv 2018', arXiv preprint arXiv:1809.02165, 2019. https://doi.org/10.48550/arXiv.1809.02165

[14] Pelletier, C., Valero, S., Inglada, J., Champion, N., Marais Sicre, C., and Dedieu, G.: 'Effect of training class label noise on classification performances for land cover mapping with satellite image time series', Remote Sensing, 2017, 9, (2), pp. 173. https://doi.org/10.3390/rs9020173

[15] Priyadharsini, R., and Sharmila, T.S.: 'Object detection in underwater acoustic images using edge based segmentation method', Procedia Computer Science, 2019, 165, pp. 759-765. https://doi.org/10.1016/j.procs.2020.01.015

[16] Ronneberger, O., Fischer, P., and Brox, T.: 'U-net: Convolutional networks for biomedical image segmentation', in Editor (Ed.)^(Eds.): 'Book U-net: Convolutional networks for biomedical image segmentation' (Springer, 2015, edn.), pp. 234-241. https://doi.org/10.1007/978-3-319-24574-4_28

[17] Strik, D.P., Domnanovich, A.M., Zani, L., Braun, R., and Holubar, P.: 'Prediction of trace compounds in biogas from anaerobic digestion using the MATLAB Neural Network Toolbox', Environmental Modelling & Software, 2005, 20, (6), pp. 803-810. https://doi.org/10.1016/j.envsoft.2004.09.006

[18] Beale, M.H., Hagan, M.T., and Demuth, H.B.: 'Neural network toolbox', User's Guide, MathWorks, 2010, 2, pp. 77-81. Corpus ID: 61288267

[19] Hossin, M., and Sulaiman, M.N.: 'A review on evaluation metrics for data classification evaluations', International journal of data mining & knowledge management process, 2015, 5, (2), pp. 1. https://doi.org/10.5121/ijdkp.2015.5201

[20] Wang, B., Li, L., Nakashima, Y., Kawasaki, R., Nagahara, H., and Yagi, Y.: 'Noisy-LSTM: Improving temporal awareness for video semantic segmentation', IEEE Access, 2021, 9, pp. 46810-46820. https://doi.org/10.1109/ACCESS.2021.3067928

[21] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. InMedical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18 2015 (pp. 234-241). Springer International Publishing. https://doi.org/10.1007/978-3-319-24574-4_28

[22] Shelhamer, E., Long, J., and Darrell, T.: 'Fully convolutional networks for semantic segmentation', IEEE transactions on pattern analysis and machine intelligence, 2017, 39, (4), pp. 640-651. https://doi.org/10.48550/arXiv.1411.4038

[23] Villamizar, M., Canevet, O., and Odobez, J.-M.: 'Multi-scale sequential network for semantic text segmentation and localization', Pattern Recognition Letters, 2020, 129, pp. 63-69. https://doi.org/10.1016/j.patrec.2019.11.001

[24] Han, C., Duan, Y., Tao, X., and Lu, J.: 'Dense convolutional networks for semantic segmentation', IEEE Access, 2019, 7, pp. 43369-43382. https://doi.org/10.1109/ACCESS.2019.2908685

[25] Gu, J.: 'Liu t., Wang X., Wang G., Cai J., Chen T', Recent Advances in Convolutional Neural Networks//Pattern Recognition, 2018, 77, pp. 354-377. https://doi.org/10.1016/j.patcog.2017.10.013

[26] Valada, A., Oliveira, G.L., Brox, T., and Burgard, W.: 'Deep multispectral semantic scene understanding of forested environments using multimodal fusion', in Editor (Ed.)^(Eds.): 'Book Deep multispectral semantic scene understanding of forested environments using multimodal fusion' (Springer, 2017, edn.), pp. 465-477. https://doi.org/10.1007/978-3-319-50115-4_41

[27] Xiang, Q., Cao, Y., Xu, H., Guo, Y., Yang, Z., Xu, L., Yuan, L., and Deng, H.: 'Identification of novel pathogenic ABCA4 variants in a Han Chinese family with Stargardt disease', Bioscience Reports, 2019, 39, (1). https://doi.org/10.1042/BSR20180872

[28] Fan, J., Zheng, P., and Shufei, L.: 'Vision-based holistic scene understanding towards proactive human-robot collaboration', Robotics and Computer-Integrated Manufacturing, 2021. https://doi.org/10.1016/j.rcim.2021.102304

[29] Boitard, R., Pourazad, M.T., and Nasiopoulos, P.: 'Compression efficiency of high dynamic range and wide color gamut pixel's representation', IEEE Transactions on Broadcasting, 2017, 64, (1), pp. 1-10. https://doi.org/10.1109/TBC.2017.2781120

[30] Kar, M.K., Nath, M.K., and Neog, D.R.: 'A review on progress in semantic image segmentation and its application to medical images', SN computer science, 2021, 2, (5), pp. 397. https://doi.org/10.1007/s42979-021-00784-5

[31] Misra, B.B., Biswal, B.N., Dash, P.K., and Panda, G.: 'Simplified polynomial neural network for classification task in data mining', in Editor (Ed.)^(Eds.): 'Book Simplified polynomial neural network for classification task in data mining' (IEEE, 2007, edn.), pp. 721-728. https://doi.org/10.1109/CEC.2007.4424542

[32] Hiew, B.Y., Tan, S.C., and Lim, W.S.: 'Intra-specific competitive co-evolutionary artificial neural network for data classification', Neurocomputing, 2016, 185, pp. 220-230. https://doi.org/10.1016/j.neucom.2015.12.051

[33] Kuznetsova, A., Rom, H., Alldrin, N. et al. The Open Images Dataset V4. Int J Comput Vis 128, 1956–1981 (2020). https://doi.org/10.1007/s11263-020-01316-z

[34] Zhu E, Ju Y, Chen Z, Liu F, Fang X. DTOF-ANN: an artificial neural network phishing detection model based on decision tree and optimal features. Applied Soft Computing. 2020 Oct 1;95:106505. https://doi.org/10.1016/j.asoc.2020.106505

[35] Pol, R.S., and Murugan, M.: 'A review on indoor human aware autonomous mobile robot navigation through a dynamic environment survey of different path planning algorithm and methods', in Editor (Ed.)^(Eds.): 'Book A review on indoor human aware autonomous mobile robot navigation through a dynamic environment survey of different path planning algorithm and methods' (IEEE, 2015, edn.), pp. 1339-1344. https://doi.org/10.1109/IIC.2015.7150956

[36] Ghali, V., and Mulaveesala, R.: 'Frequency modulated thermal wave imaging techniques for non-destructive testing', Insight-Non-

Destructive Testing and Condition Monitoring, 2010, 52, (9), pp. 475-480. https://doi.org/10.1784/insi.2010.52.9.475

[37] Cilimkovic M. Neural networks and back propagation algorithm. Institute of Technology Blanchardstown, Blanchardstown Road North Dublin. 2015;15(1). Corpus ID: 18592533

[38] Sosa, R., and Sosa, W.: 'A non-parametric mathematical model to investigate the dynamic of a pandemic', MedRxiv, 2020, pp. 2020.2004. 2030.20086199. https://doi.org/10.1101/2020.04.30.20086199

[39] Sarkar, K., Nasipuri, M., and Ghose, S.: 'Machine learning based keyphrase extraction: comparing decision trees, naïve Bayes, and artificial neural networks', Journal of Information Processing Systems, 2012, 8, (4), pp. 693-712. https://doi.org/10.3745/JIPS.2012.8.4.693

[40] Carrizosa Priego, E.J., Molero Río, C., and Romero Morales, M.D.: 'Mathematical optimization in classifcation and regression trees', 29, 6-33., 2021. https://doi.org/10.1007/s11750-021-00594-1

[41] Hilu-Dadia, R., Hakim-Mishnaevski, K., Levy-Adam, F., and Kurant, E.: 'Draper-mediated JNK signaling is required for glial phagocytosis of apoptotic neurons during Drosophila metamorphosis', Glia, 2018, 66, (7), pp. 1520-1532. https://doi.org/10.1002/glia.23322

[42] Tan, J., Yang, J., Wu, S., Chen, G., and Zhao, J.: 'A critical look at the current train/test split in machine learning', arXiv preprint arXiv:2106.04525, 2021. https://doi.org/10.48550/arXiv.2106.04525

[43] Zafer, C.: 'Fusing fine-tuned deep features for recognizing different tympanic membranes', Biocybernetics and Biomedical Engineering, 2020, 40, (1), pp. 40-51. https://doi.org/10.1016/j.bbe.2019.11.001

[44] Parkhi, O. M., Vedaldi, A., Zisserman, A., & Jawahar, C. V. (2012). Cats and dogs. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 3498–3505. https:// doi.org/10.1109/CVPR.2012.6248092. https://doi.org/10.1109/CVPR.2012.6248092

[45] Krizhevsky, A., Sutskever, I., and Hinton, G.E.: 'Imagenet classification with deep convolutional neural networks', Communications of the ACM, 2017, 60, (6), pp. 84-90. https://doi.org/10.1145/3065386

[46] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L.: 'Imagenet: A large-scale hierarchical image database', in Editor (Ed.)^(Eds.): 'Book Imagenet: A large-scale hierarchical image database' (Ieee, 2009, edn.), pp. 248-255. https://doi.org/10.1109/CVPR.2009.5206848

[47] Demuth, H.B., Beale, M.H., De Jess, O., and Hagan, M.T.: 'Neural network design' (Martin Hagan, 2014). http://thuvienso.thanglong.edu.vn//handle/TLU/5760

[48] Svozil, D., Kvasnicka, V., and Pospichal, J.: 'Introduction to multi-layer feed-forward neural networks', Chemometrics and intelligent laboratory systems, 1997, 39, (1), pp. 43-62. https://doi.org/10.1016/S0169-7439(97)00061-0

[49] Dong, BX., Shan, M. & Hwang, BG. Simulation of transportation infrastructures resilience: a comprehensive review. Environ Sci Pollut Res 29, 12965–12983 (2022). https://doi.org/10.1007/s11356-021-18033-w

[50] Chollet, F.: 'Deep learning with Python' (Simon and Schuster, 2021. 2021). ISBN 9781617294433

[51] Byrd, R.H., Lu, P., Nocedal, J., and Zhu, C.: 'A limited memory algorithm for bound constrained optimization', SIAM Journal on scientific computing, 1995, 16, (5), pp. 1190-1208. https://doi.org/10.1137/0916069

[52] Hager, W.W., and Zhang, H.: 'Algorithm 851: CG_DESCENT, a conjugate gradient method with guaranteed descent', ACM Transactions on Mathematical Software (TOMS), 2006, 32, (1), pp. 113-137. https://doi.org/10.1145/1132973.1132979

[53] Haykin, S.: 'Self-organizing maps', Neural networks-A comprehensive foundation, 2nd edition, Prentice-Hall, 1999. ISBN:978-0-13-273350-2

[54] Riedmiller M, Braun H. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. InIEEE international conference on neural networks 1993 Mar 28 (pp. 586-591). IEEE. http://dx.doi.org/10.1109/ICNN.1993.298623

[55] Hagan, M.T., and Menhaj, M.B.: 'Training feedforward networks with the Marquardt algorithm', IEEE transactions on Neural Networks, 1994, 5, (6), pp. 989-993. https://doi.org/10.1109/72.329697

[56] Mollor, M.: 'A scaled conjugate gradient algorithm for fast supervised learning [J]', Neural networks, 1993, 6, (4), pp. 525-534. https://doi.org/10.1016/S0893-6080(05)80056-5

[57] Howard, D., and Mark, B.: 'MATLAB neural network toolbox documentation', The MathWorks, 2004. www.mathworks.com

تقييم أداء طرق الشبكة العصبية الاصطناعية بناءً على تصنيف التعلم الآلي للكتل

[1]ريا أكرم حمدي [2]محمد جاجان يونس

[1،2]جامعة الموصل – كلية علوم الحاسوب والرياضيات – قسم علوم الحاسوب

Email: mohammed.c.y@uomosul.edu.iq

تاريخ الاستلام: 31/7/2023   تاريخ القبول: 13/9/2023

الملخص

تستغرق تقنيات التصنيف التقليدية (كل بكسل على حدا) وقتاً طويلاً في التنفيذ، في حين أن التجزئة الدلالية في التعلم الآلي تتطلب تتحديد الحقائق الأرضية لكل فئة في بكسلات الصورة. تقترح هذه الدراسة طريقة التجزئة الدلالية من خلال تجزئة الصورة الى أجزاء أصغر (5x5) تسمى الكتلة، ثم يتم استخراج الميزات أو الخصائص الرئيسية وتدريب النموذج على تلك الميزات باستخدام خصائص الالوان والقوام لتلك الكتل. استخدمت هذه الدراسة قاعدة بيانات Oxford-IIIT Pet، وأُختيرت ثلاثين صورة للقطط من مجموعة البيانات هذه. دُربت وقُيمت خمسة نماذج مختلفة للشبكات العصبية الاصطناعية لكل من الأسلوبين القائمين على البكسل وعلى الكتلة: نموذج الانتشار العكسي لفنبرك ماركورت، نموذج برويدن فليجر كولدفرب شانو، نموذج الانتشار الخلفي المرن، نموذج الانتشار الخلفي المتدرج المتقارب ونموذج الانتشار الرجعي لمعدل التعلم المتغير. أظهرت النتائج أن الدقة المستندة إلى البكسل تتراوح من 70.82٪ إلى 76.47٪ ، بينما تتراوح الدقة القائمة على الكتلة من 82.94٪ إلى 85.83٪. ويُلاحظ ايضاً أن وقت المعالجة للنهج المستند إلى البكسل يستغرق وقت أطول من الأسلوب المستند إلى الكتلة، إذ يستغرق نموذج الانتشار الخلفي المرن للنهج القائم على البكسل 242.39 ثانية، بينما يستغرق نموذج الانتشار الرجعي لمعدل التعلم المتغير أقصر وقت معالجة 49.89 ثانية. أما بالنسبة للنهج القائم على الكتلة، فيستغرق نموذج الانتشار العكسي لفنبرك ماركورت أطول وقت معالجة 13.86 ثانية، في حين لا يزال نموذج الانتشار الرجعي لمعدل التعلم المتغير أقصر وقت معالجة كما هو الحال في نهج البكسل 5.98 ثانية. لذلك، تُعد الطرائق القائمة على الكتلة أكثر كفاءة فيما يتعلق بالوقت والدقة لنماذج التصنيف. وقد أوضحت نتائج الاختبار أن دقة نموذج الانتشار العكسي لفنبرك ماركورت هي الأعلى لصور الاختبار الثلاثة والتي تراوحت بين 94.72٪ الى 89.81٪، بينما نموذج الانتشار الرجعي لمعدل التعلم المتغير قد أعطى أقل دقة والتي تراوحت بين 92.96٪ إلى 81.15٪. وأما النماذج المتبقية فقد أعطت مستويات متوسطة من الدقة.

الكلمات المفتاحية : الشبكة العصبية الاصطناعية، التجزئة الدلالية، استخراج الميزات.