Journal of Soft Computing and Computer Applications

Manuscript 1010

Enhancing Image Classification Using a Convolutional Neural Network Model

Zena M. Saadi

Ahmed T. Sadiq

Omar Z. Akif

Marwa M. Eid

JOURNAL OF SOFT COMPUTING AND COMPUTER APPLICATIONS 2024;1:1010

ORIGINAL STUDY

Enhancing Image Classification Using a Convolutional Neural Network Model

Zena M. Saadi [a,*], Ahmed T. Sadiq [a], Omar Z. Akif [b], Marwa M. Eid [c]

[a] Department of Computer Science, University of Technology – Iraq, Al-Sina'a St., Al-Wehda District, 10066 Baghdad, Iraq
[b] Department of Computer Science, College of Education for Pure Science (Ibn al-Haitham), University of Baghdad, Adhamiya District, Antar Square, 10071 Baghdad, Iraq
[c] Faculty of Artificial Intelligence, Delta University for Science and Technology, Mansoura, Egypt, Middle East University, MEU Research Unit, 11831 Amman, Jordan

ABSTRACT

In recent years, with the rapid development of the current classification system in digital content identification, automatic classification of images has become the most challenging task in the field of computer vision. As can be seen, vision is quite challenging for a system to automatically understand and analyze images, as compared to the vision of humans. Some research papers have been done to address the issue in the low-level current classification system, but the output was restricted only to basic image features. However, similarly, the approaches fail to accurately classify images. For the results expected in this field, such as computer vision, this study proposes a deep learning approach that utilizes a deep learning algorithm. In this research, a proposed model based on a Convolutional Neural Network (CNN) which is a machine learning tool that can be used for the automatic classification of images. The model is concerned with the classification of images, and for this, it employs the COREL Image dataset (Corel Gallery Image Dataset) as a reference. The images in the dataset used for training are harder than the classification of the images since they need more computational resources. In the experimental part, training the images using the CNN network achieved 98.52% accuracy, proving that the model has high accuracy in the classification of images.

Keywords: Convolutional neural network, Images classification, COREL image dataset, Deep learning

1. Introduction

These days, the Internet contains an overwhelming number of images and videos, which stimulates the creation of search applications and algorithms studying the semantic

Received 26 August 2024; accepted 27 October 2024.
Available online 31 December 2024

* Corresponding author.
E-mail addresses: cs.22.15@grad.uotechnology.edu.iq (Z. M. Saadi), ahmed.t.sadiq@uotechnology.edu.iq (A. T. Sadiq), omar.z.a@ihcoedu.uobaghdad.edu.iq (O. Z. Akif), mmm@ieee.org (M. M. Eid).

https://doi.org/10.70403/3008-1084.1010

requirements of images and videos to provide the user with richer search content and summaries [1]. Many researchers around the world have made many innovations in the field of image labeling. This makes formulating approaches concerning object detection and scene classification problems possible. Since then, artificial neural networks have provided a performance breakthrough, especially convolutional neural networks [2].

Image classification is one of the most important research areas of computer vision and the foundation of numerous categories of vision recognition. Enhancing the performance of a classification network usually greatly enhances its application level, such as object detection, segmentation, human pose estimation, video classification, object tracking, and super-resolution technology [3]. Strengthening the ability in image classification is greatly significant in advancing the cause of computer vision. Its principal activity encompasses image data preprocessing, feature extraction and feature representation, and classifier construction [4].

Most of the current traditional learning techniques (multilayer perception machine, support vector machine, etc.) are mainly based on shallow architectures to handle limited sample cases and computational power. However, for target objects with rich meaning, the performance and the generalization ability in the complex classification problem are insufficient [5]. Because of CNN's success in solving image classification and recognition problems, CNN has almost successfully solved many image processing problems and greatly improved many machine learning problems. It has now evolved into a flexible and general deep-learning architecture [6].

Image classification's main research emphasis has always been on feature extraction of images to classify images. The most traditional techniques for image feature extraction focus on the automatic definition of specific image features. It has low generalization performance. Thus, letting a computer upload the possibility of processing images in a way that is close to biological vision is what researchers want [7]. Artificial Neural Network (ANN) is an abstract biological neural network, which is a mathematical operation model constituted by numerous neurons with a complex connection mode. It tends to mimic the neural network functionality involved in the evaluation of neural signals [8].

The contribution of the proposed model is the creation of a Convolutional Neural Network (CNN), which is a machine learning that enables easy classification of images. To this effect, the tools rely on the COREL image dataset. The training images are more difficult than the classification of the images owing to the increase in the computational intensity.

This research is organized as follows: the latest studies that have been used in image classification methods are presented in Section 2. Section 3 briefly explains the materials and methods used in this research. A comparison with previous studies is introduced in Section 3.10. Discussion is given in Section 4. Conclusions and future work are given in Section 5.

2. Related work

Kuo *et al.* [9] explained that incorporating deep convolutional features enhances the classification capacity of images and, hence, the relevance of the image retrieval results. This approach can be regarded as a transition from the approach based on feature engineering, which is initially static, to a learning-based approach that exhibits greater flexibility in the case of large, evolving image datasets.

The general framework of the Convolutional Neural Network (CNN) proposed by Hsin et al. [10] focused on ensemble models for aggregate image retrieval. The authors used two efficient deep learning networks as an image classifier that is required to extract image features. These networks were AlexNet and Network In Network (NIN).

Medus *et al.* [11] applied CNN to the classification of hyperspectral images, with an application in the industrial food packaging domain. Hyperspectral imaging captures and analyzes the data in the full spectrum range and discriminates the objects by their spectral responses. This can be important for maintaining product quality, sorting parts for recycling, and following food hygiene standards.

Al-Saffar *et al.* [12] showed that the classification framework is called the region-based pluralistic CNN, and it uses context-aware semantic representations to get useful features. Fusing several discriminant appearance factors as vector with the CNN enables to show the spatial-spectral contextual sensitivity as the need for accurate pixel classification. Concerning the proposed method, it is assumed that contextual features of interaction learning through different inputs at the region level improve discriminating ability. Therefore, subsequent to their fusion, both the spectral-spatial features are fed into a fully connected network, with pixel vector label predictions obtained through the Softmax layer. Based on the experimental comparison of the results of hyperspectral image datasets that are generally used in the present study, it can be found that the proposed technique is significantly more effective than other traditional deep-learning-based classifiers and many other advanced classifiers.

Yan *et al.* [13] suggested a Spatial and Class Structure Regularized Sparse Representation (SCSSR) graph or graph for semi-supervised HyperSpectral Image (HSI) classification. In particular, graph Laplace was introduced into the Spectral Regression (SR) model, that is, spatial neighbors should have similar representation coefficients in the SR model, and thus, the obtained coefficients matrix is more reflective of the similarity between samples. Moreover, a probabilistic class is introduced into the SR model; the probabilistic that each sample belongs to every class is used to further the contrast in the graph.

According to Zhang *et al.* [14], rotation invariance and uniform sample specificity are especially significant for object detection and classification applications. The information that distinguishes their research from previous work is studying the specific types of invariances, for example, rotation invariance. As for the object classification in this research, a novel multichannel CNN (mCNN) is developed to identify the invariant features. To alleviate the change of the characteristic variance of two samples in the same class and with diverse rotation, multi-channel convolution sharing the same weight is employed. Thus, the uniform object meets the invariance and the rotation invariance is met at the same time to enhance the invariance of the feature.

Most importantly, the proposed mCNN is more advantageous when the training samples are small. In conclusion, for two datasets of handwriting recognition, the experiment result shows that the proposed mCNN is more suitable for learning the features that are highly invariant when the number of training samples is few. With the development of big data and more hidden layers, CNN has a more complex network structure that can become feature learning, and then the original machine learning methods have shortcomings in certain aspects of their ability to express features. This is the case because the advanced model known as the CNN trained using the deep learning algorithm has offered a solution to numerous large-scale recognition issues in the computer vision field.

3. Materials and methods

3.1. The COREL image dataset

Several high-quality digital images are selected from the COREL Images database which constitutes only a small fraction of the total image database used to train machine learning algorithms and computer vision systems. Therefore, this dataset is becoming more

Fig. 1. Some images are in a subset of the COREL image dataset [16].

and more popular for machine learning tasks. The COREL Images dataset has a diverse range of images, covering various topics, such as landscapes, cityscapes, people, portraits, animals, wildlife, food, drink, and objects, all of which are 32×32 in size and are Joint Photographic Experts Group (JPEG) and Tag Image File Format (TIFF) formats, with a resolution of 256×384 pixels or 384×256 pixels and a maximum file size of 10 MB. So, there are 10 concept groups of images composed of 100 images. For each concept group, the images are divided into 70 images for training, 10 for validation, and 20 for testing and evaluating the proposed model, as illustrated in Fig. 1, which shows the sample images from each semantic category from the COREL 1000 image dataset [15].

3.2. Convolutional neural network

In deep learning, the network seeks to identify complex features from the data with high dimensions and then use these features to build a model that maps input data to the output data (like classes) [17]. These deep learning architectures are commonly developed as neuron networks, where higher-level properties are estimated like nonlinear functions of lower-level properties [18]. An example of a deep learning structure is the layer neural network, which is the most common in use. In its hierarchical structure, deep learning can be divided into many layers. The following section will describe these layers [19].

CNN is a type of deep learning neural network employed for structures with data arranged in a grid format, like images, whose purpose is to find the attributes that are learned from the database. The convolutional layer will be of size $N \times N$ and, in this layer, the dot product multiplication will be performed between the neighborhood values. Therefore, the convolutional layer only includes addition and multiplication operations in its formulation, as shown in Fig. 2 [17].

The key features of CNNs include:

1. **Convolutional Layers**: These layers convolve (or correlate) filters (or kernels) with the input data to find patterns such as edges and texture. This operation helps to decrease the number of dimensions with structure retention of essential characteris-

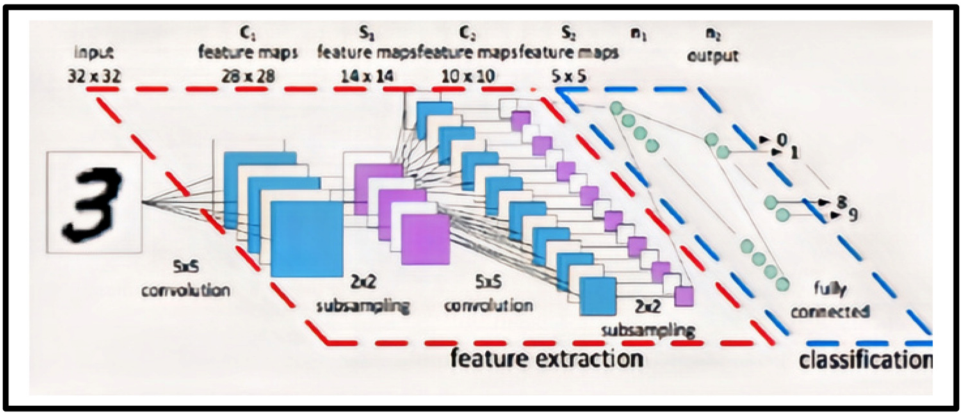


Fig. 2. CNN's internal layers [20].

tics. The filters are those small matrices that move over the image and carry out dot product at every point. It proceeds to emphasize particular patterns in the image.
2. **Pooling Layers**: Downsampling layers in a pooling layer decrease the size of the feature maps that are generated by the convolutional layers and decrease the parameter and computational complexity of the network to avoid overfitting. Some of them are max pooling, where the maximum value under the window is considered, and average pooling, where the average of the values in the window is considered.
3. **Activation Functions**: After convolution, a non-linear activation function is applied, such as the Rectified Linear Unit (ReLU) to enable the network to learn non-linear features.
4. **Fully Connected Layers**: At the end of the network, these layers link each neuron from the previous layer to all neurons in the current layer, typically for classification purposes.

CNNs are well suited for image recognition, object detection, as well as numerous other computer vision tasks since they can learn features of raw data on their own [21–23].

3.3. Generation convolutional neural network model phase

At this phase, many procedures will be clarified in the proposed model as it will play a crucial role in establishing the classification model. The CNN structure of the proposed model has 10 layers, as illustrated in Fig. 3.

3.4. Proposed model

The structure of the proposed CNN shows the sequence of the operations from the input to the output, as shown in Fig. 3, and is explained as follows:

1. **Input Layer:** This layer takes the input image that is usually represented as a multi-dimensional array, with dimensions corresponding to height, width, and red, green, and blue color channels.
2. **Convolutional Blocks:** These blocks are the core of CNNs, responsible for extracting features from the input image. Each block typically contains: Convolutional Layers, Activation Functions and Pooling Layers, as shown in the following paragraphs.

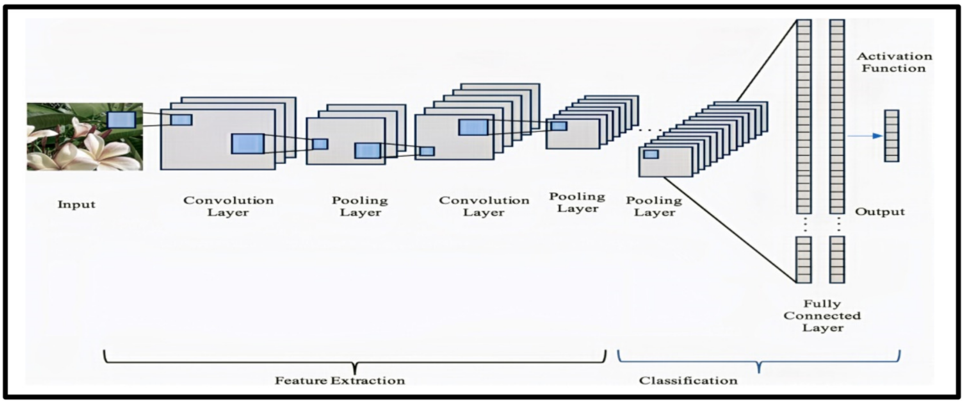


Fig. 3. The proposed CNN model for classification images.

a. Block 1 (Conv)

- Block 1 - Conv1: The first convolution in the block applies a kernel of size (3 × 3) with a stride of 1, using 64 filters. This extracts basic features from the input image.
- Block 1 - Conv2: The second convolution uses a kernel of (3 × 3) with a stride of 1, employing 64 filters. It builds upon the features extracted in the previous layer.
- Block 1 - Pool: A max pooling layer with a window size of (2 × 2) and a stride of 2 is applied, downsampling the feature maps.

b. Block 2 (Conv)

- Block 2 - Conv1: Similar to the first block, this convolution uses a kernel of size (3 × 3) with a stride of 1, employing 128 filters.
- Block 2 - Conv2: Another convolution with a (3 × 3) kernel, stride of 1, and 128 filters.
- Block 2 - Pool: Max pooling with a window size of (2 × 2) and stride of 2, further reducing the dimensions.

c. Block 3 (Conv)

- Block 3 – Conv1: A (3 × 3) kernel with stride 1 and 256 filters.
- Block 3 – Conv2: A (3 × 3) kernel with stride 1 and 256 filters.
- Block 3 – Conv3: A (3 × 3) kernel with stride 1 and 256 filters.
- Block 3 – Conv4: A (3 × 3) kernel with stride 1 and 256 filters.
- Block 3 – Pool: Max pooling with a window size of (2 × 2) and stride of 2.

d. Block 4 (Conv)

- Block 4 – Conv1: A (3 × 3) kernel with stride 1 and 512 filters.
- Block 4 – Conv2: A (3 × 3) kernel with stride 1 and 512 filters.
- Block 4 – Conv3: A (3 × 3) kernel with stride 1 and 512 filters.
- Block 4 – Conv4: A (3 × 3) kernel with stride 1 and 512 filters.
- Block 4 – Pool: Max pooling with a window size of (2 × 2) and stride of 2.

e. Block 5 (Conv)

- Block 5 – Conv1: A (3 × 3) kernel with stride 1 and 512 filters.
- Block 5 – Conv2: A (3 × 3) kernel with stride 1 and 512 filters.
- Block 5 – Conv3: A (3 × 3) kernel with stride 1 and 512 filters.
- Block 5 – Conv4: A (3 × 3) kernel with stride 1 and 512 filters.
- Block 5 – Pool: Max pooling with a window size of (2 × 2) and stride of 2.

3. **Global Max Pooling:** This layer aggregates the spatial information from the last convolutional block, reducing the feature maps to a single value per feature. It takes the maximum value within each feature map, summarizing the key information learned by the network in the proposal structure using the five pooling and one max pooling function.
4. **Flattening:** The output of the global max pooling is flattened, converting it into a one-dimensional vector. This is necessary for feeding the data into the fully connected layers.
5. **Fully Connected Layers:** These layers perform the classification task by taking the flattened feature vector and processing it to predict the class labels. These layers are:
   - Dense Layers: These layers apply a linear transformation to the input, followed by an activation function in the proposal structure for CNN using two dense function layers.
   - Dropout Layer: This layer randomly disables a fraction of neurons during training, preventing overfitting.
6. **Output Layer:** The final layer of the network is the output layer that predicts class probabilities. The activation function used here is the softmax function which ensures that the output probabilities for all classes sum to 1.

Fig. 4 breaks down the proposed CNN into its layers:

3.5. Parameters of the proposed model

The proposed model is used to expect and extract high-level features. To improve the efficiency of the proposed model, many parameters have been used here, as depicted in Table 1.

3.6. Feature extraction phase

Here, the CNN is suggested to get features from models by using model's output flattening. The flattening converts the received 2D matrices into a vector. For each of the

Table 1. CNN algorithm parameters.

Parameter name	Parameter value	Description
Kernal	3×3	This filter is used to extract features from images.
Padding	1	Determines how the input is padded before convolution (often "same" to maintain the same spatial dimensions).
Dropout	25%	Randomly drops out a portion of input neurons to zero during training to prevent overfitting.
Pooling (Max pooling)	2×2	Selects the maximum value from a defined region of the feature map. This retains important features while reducing dimensionality.
Epochs	10	Ten times the neural network will be used to process the COREL images dataset.
batch size	32	The number of input data samples that the model takes as input in one single update.
weight decay	0.0001	Used to prevent overfitting by adding a penalty term to the loss function for large weight values.
Loss function	Categorical cross-entropy	To quantify the error between the predicted output and the desired output, so that the model can learn to minimize this error during training.

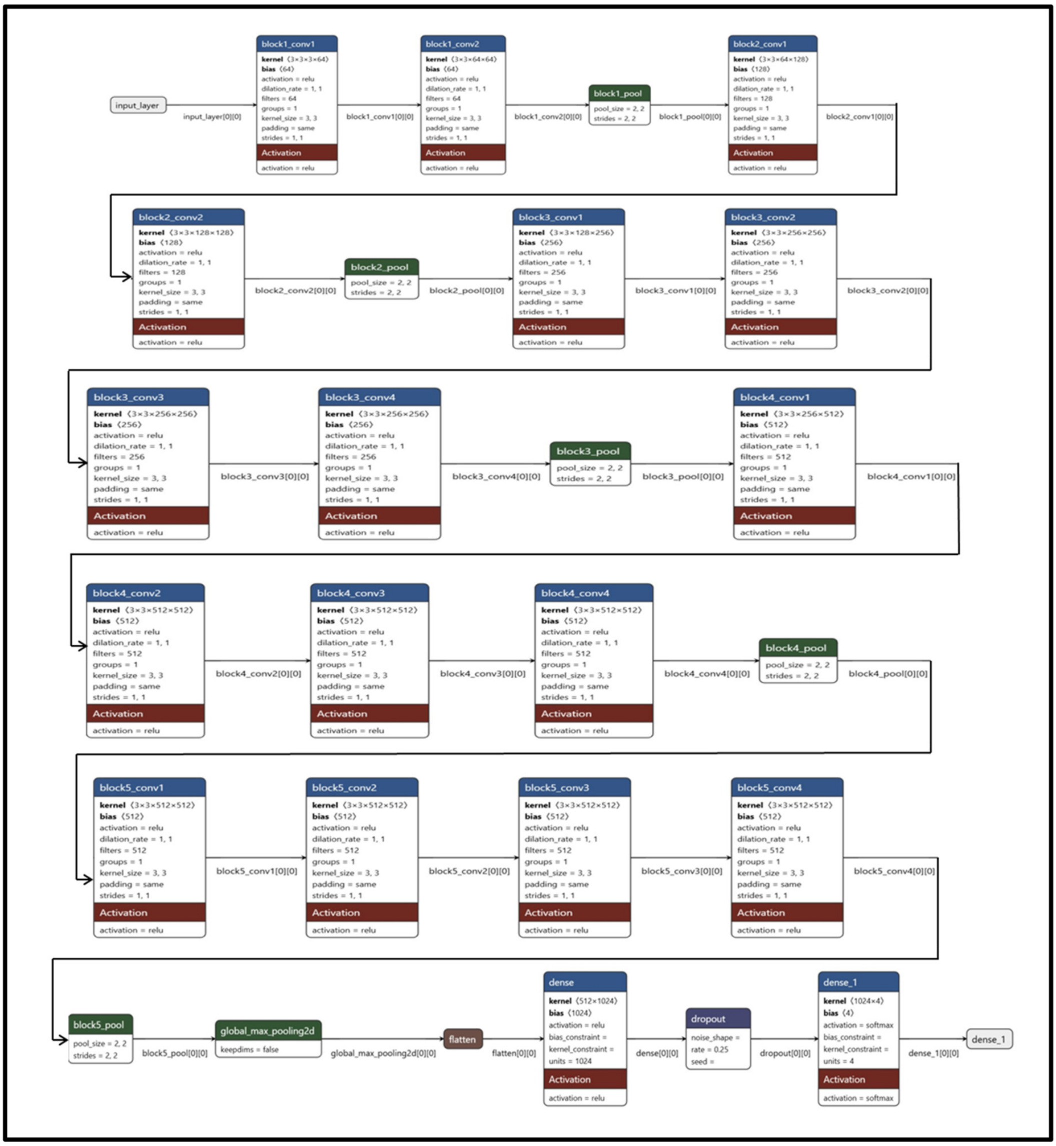
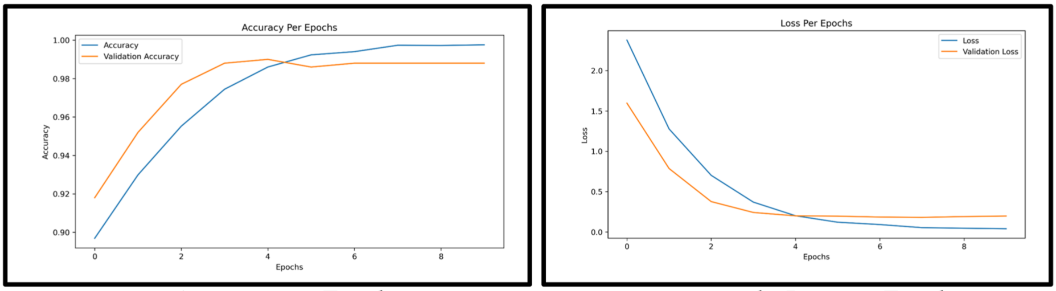


Fig. 4. The proposed CNN architecture is drawn using the Netron software.

training images, it generates 1024 features and provides the feature vector for all the training images in the form of a matrix.

3.7. Convolutional neural network layers

Six layers based on CNN were used in the proposed model. This means that every training images subset is passed via all these six layers. The network will be a fully connected layer, five maximum pool layers, and four convolution layers. Fig. 3 shows the input images that the network layers have changed.



a- Accuracy per Epoch b- Loss per Epoch

Fig. 5. Proposed model analysis.

3.8. Model analysis

Many of the assumptions of machine learning are based on visualizations. Both professionals and scholars often use visualization to control the learned parameters and the output results to enhance their models. Also, there is a module to display the distribution of tensors images and sounds in the TensorBoard component of the dashboard. In this research, *loss, and accuracy* changes have been demonstrated over each epoch, as displayed in Fig. 5. Feed-forward and back-propagation are used in neural networks to process the complete dataset. In this case, it is crucial to comprehend the meaning of loss and accuracy, like training proceeds at which time these are constant. By understanding this scaler graph, there will be less overfitting for this reason. The above information explains and describes the scaler graph and how to avoid overfitting. Accordingly, the researchers observed a clear correlation between the increase in losses and accuracy in the training and validation samples. Based on this indicator, the proposed model in this research does not suffer from the issue of overfitting.

3.9. Training results

The model has been trained with 700 training images (1000 red, 1000 green, 1000 blue) using the optimization method Stochastic Gradient Descent (SGD) with an initial learning rate of 0.0001. To optimize the images, the dataset is passed through 10 epochs. In each epoch, the weights are adjusted in order to have an image that is closer to the desired image. Table 2 shows the training evaluation of the model's accuracy and loss, along with the associated hyperparameters. As shown in this table, the number of iterations both for the training and validation process for images was done over 32 Epochs, where 700, 100, and 200 were in every 32 used. First, the initial learning rate of 0.0001 was assigned in the beginning of the training phase. During several Epochs, the learning rate was increased up to 0.006, so the system increases its performance according to the data that is used for training.

Table 2. Training results.

		Learning rate													Total parameters	
Iteration	Batch size	Initial value	Last value	Loss function	Optimizer	Epochs	ETA	VAL-Loss	VAL-Accuracy	Loss	F1_Score	Precision	Recall	Accuracy	Learning learing	Non-learning
32	32	0.0001	0.006	Categorical cross entropy	SGA	10	16.749 MS	0.1213	0.9880	0.0420	0.9882	0.9888	0.9876	0.9976	20,553,796	0

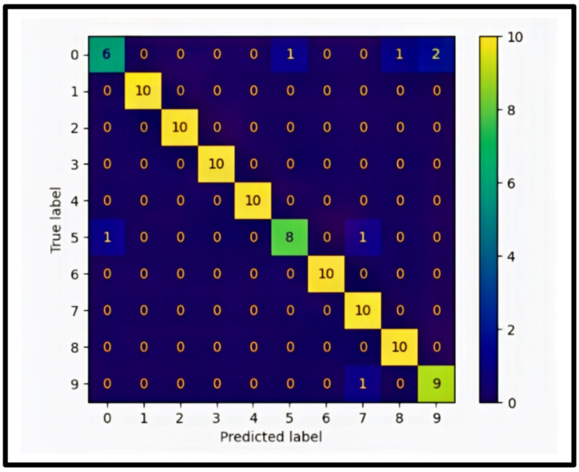


Fig. 6. Confusion matrix.

3.10. Testing results

As mentioned before, 100 testing images out of a total of 1000 images in the COREL image dataset consist of 10% of the database that has been used in the input into the model architecture and has passed through all the layers of the model. In testing, the saved parameters and weights that the network reached were used, then the test images were multiplied by the weights, and then the images were classified into 10 classes. The confusion matrix in test model proposal is presented in Fig. 6.

Here are some key metrics that can be derived from a Confusion Matrix [24]:

- Accuracy = (TP + TN) / (TP + TN + FP + FN) - This measures the overall accuracy of the model.
- Precision = TP / (TP + FP) - This measures the proportion of true positives among all predicted positives.
- Recall = TP / (TP + FN) - This measures the proportion of true positives among all actual positives.
- F1-score = 2 × (Precision × Recall) / (Precision + Recall) - This is the harmonic mean of precision and recall.
- False Positive Rate = FP / (FP + TN) - This measures the proportion of false positives among all actual negatives.
- False Negative Rate = FN / (FN + TP) - This measures the proportion of false negatives among all actual positives.

where TP is the True Positive, TN is the True Negative, FP is the False Positive, and FN is the False Negative.

Therefore, the above rules can be used to find the measure test for the proposed model in Table 3.

Table 3. Performance of testing.

Loss	F1_Score	Accuracy	Precision	Recall
0.0842%	98.82%	98.52%	93.67%	91.37%

Table 4. Image classification accuracy on the COREL image datasets.

Related works	Image classification accuracy
Kuo *et al.* [9]	96.9%
Hsin *et al.* [10]	90.19%
Medus *et al.* [11]	94.0%
M. Manikandakumar and P. Karthikeyan [25]	97.79%
Hong *et al.* [26]	98.40%
The proposed model	98.52%

3.11. Timing test

The time result of the proposed model (in seconds) for training is 16.749.234. All presented trials were conducted on a computer with a Core i7 processor, a clock rate of up to 2.7 GHz, and 16GB of memory with a preloaded Windows 10 operating system.

4. Comparison with previous researches

Many methods are suggested in the related works to enhance the classification performance of images. Table 4 displays the summary of image classification accuracy for the COREL image datasets from various researches.

From this table, researchers will thus be able to deduce that the proposed model is superior as its results outperforms that of the previous studies.

5. Discussion

The related works discussed previously included outcomes that were relatively lower than the results of this research. The results set out herein show that the CNN that was developed in the context of this research did perform better than other research in the classification process. This research also observed that the flattening layer, for example, had features that were used by classifying similar images, because this layer had properties that depended on trained weights during the training phase. Also, the proposed model increased the dataset samples, overcoming overfitting issues to enhance training accuracy.

6. Conclusions

This research has provided a proper image classification model that can semantically categorize the right images with high classification efficiency. Several algorithms were employed at once to work towards the best outcomes. In creating the proposed model to optimize the features, the hamming distance approach is utilized to measure the distance between the kept and input factors. It also eliminates the overfitting problem and, at the same time, enhances the model's performance by increasing the number of training images.

The contribution presented is that the CNN model has a very high accuracy of classification that enabled the extraction of the most significant features of the images to a level of 98.52%. This entails the process of escalating the number of samples of the dataset by including expected effects on images and choosing the finest features from the flattened layer, which significantly leads to high classification accuracy. These features are utilized to find the distance between each kept image feature and the input image features, which enhances the image classifying accuracy up to a certain extent and also minimizes the time of image classifying. The proposed model in CNN is highly effective in image classification recognition tasks, as it can learn hierarchical features from the input image. CNNs are widely used in applications, such as image recognition, and object detection.

There are numerous practical applications for this proposal, including creating a safe search engine and the health sector, as this is a proposal that requires safe protocol when it comes to patient-to-hospital communication. One of the future works is to develop a model for ANN to use in tasks like video monitoring or diagnosing utilities.

Acknowledgment

I would like to express my deep gratitude to Professor Dr. Alaa Kadhim Farhan, head of Computer Science Department, whose expertise and knowledge were invaluable during this research.

Authors' contributions

Conceptualization, Z.M.S. and A.T.S.; methodology, O.Z.A.; software, M. M. E.; validation, Z.M.S., A.T.S. and O.Z.A.; formal analysis, M. M. E.; investigation, O.Z.A.; resources, A.T.S.; data curation, Z.M.S.; writing—original draft preparation, Z.M.S.; writing—review and editing, O.Z.A.; visualization, M. M. E.; supervision, O.Z.A.; project administration, A.T.S.; funding acquisition, M. M. E. All authors have read and agreed to the published version of the manuscript.

Conflicts of interest

The authors declare no conflict of interest.

Data availability

No External Data.

References

1. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. of the 2016 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp. 2818-2826, 27–30 June 2016, doi: 10.1109/CVPR.2016.308.
2. B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Object detectors emerge in deep scene CNNs," 2015, *arXiv:1412.6856*.
3. A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, P. Martinez-Gonzalez, and J. Garcia-Rodriguez, "A survey on deep learning techniques for image and video semantic segmentation". *Applied Soft Computing*, vol. 70, pp. 41–65, 2018, doi: 10.1016/j.asoc.2018.05.018.
4. Y. Chen, Y. Tian, and M. He, "Monocular human pose estimation: A survey of deep learning-based methods," *Computer Vision and Image Understanding*, vol. 192, Art. no. 102897, March 2020, doi: 10.1016/j.cviu.2019.102897.
5. H. Ahn and C. Yim, "Convolutional neural networks using skip connections with layer groups for super-resolution image reconstruction based on deep learning," *Applied Sciences*, vol. 10, no. 6, Art. no. 1959, March 2020, doi: 10.3390/app10061959.
6. D. Ciregan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Proc. of the 2012 IEEE Conf. on Computer Vision and Pattern Recognition*, Providence, RI, USA, pp. 3642–3649, 16–21 June 2012, doi: 10.1109/CVPR.2012.6248110.

7. A. Krizhevsky, I. Sutskever, and G. Hinton, “ImageNet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017, doi: 10.1145/3065386.
8. D. Zhang, B. Liu, C. Sun, and X. Wang, “Learning the classifier combination for image classification,” *Journal of Computers*, vol. 6, no. 8, pp. 1756–1763, August 2011, doi: 10.4304/jcp.6.8.1756-1763.
9. C.-H. Kuo, Y.-H. Chou, and P. Chang, “Using deep convolutional neural networks for image retrieval,” *Electronic Imaging*, vol. 28, Art. no. art00014, February 2016, doi: 10.2352/ISSN.2470-1173.2016.2.VIPC-231.
10. H.-K. Huang, C.-F. Chiu, C.-H. Kuo, Y.-C. Wu, N. N. Y. Chu, and P.-C. Chang, “Mixture of deep CNN-based ensemble model for image retrieval,” in *Proc. of the 2016 IEEE 5thGlobal Conf. on Consumer Electronics*, Kyoto, Japan, pp. 1-2, 11–14 October 2016, doi: 10.1109/GCCE.2016.7800375.
11. L. D. Medus, M. Saban, J. V. Francés-Víllora, M. Bataller-Mompeán, and A. Rosado-Muñoz, “Hyperspectral image classification using CNN: Application to industrial food packaging,” *Food Control*, vol. 125, Art. no. 107962, July 2021, doi: 10.1016/j.foodcont.2021.107962.
12. S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, “1D convolutional neural networks and applications: A survey,” *Mechanical Systems and Signal Processing*, vol. 151, Art. no. 107398, April 2021, doi: 10.1016/j.ymssp.2020.107398.
13. X. Zhang *et al.*, “Hierarchical bilinear convolutional neural network for image classification,” *IET Computer Vision*, vol. 15, no. 3, pp. 197–207, March 2021, doi: 10.1049/cvi2.12023.
14. C. Zhang *et al.*, “A hybrid MLP-CNN classifier for very fine resolution remotely sensed image classification,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 140, pp. 133–144, June 2018, doi: 10.1016/j.isprsjprs.2017.07.014.
15. A. Shabbir *et al.*, “Satellite and scene image classification based on transfer learning and fine tuning of RESNET50,” *Mathematical Problems in Engineering*, vol. 2021, no. 1, Art. no. 5843816, July 2021, doi: 10.1155/2021/5843816.
16. J. Li and J. Z. Wang, “Real-time computerized annotation of pictures,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 6, pp. 985–1002, June 2008, doi: 10.1109/TPAMI.2007.70847.
17. E. A. Alsaedi and A. K. Farhan, “Retrieving encrypted images using convolution neural network and fully homomorphic encryption,” *Baghdad Science Journal*, vol. 20, no. 1, pp. 206–220, February 2023, doi: 10.21123/bsj.2022.6550.
18. Q. Zhang, M. Zhang, T. Chen, Z. Sun, Y. Ma, and B. Yu, “Recent advances in convolutional neural network acceleration,” *Neurocomputing*, vol. 323, pp. 37–51, January 2019, doi: 10.1016/j.neucom.2018.09.038.
19. C. Lu, M. Kozakai, and L. Jing, “Sign language recognition with multimodal sensors and deep learning methods,” *Electronics*, vol. 12, no. 23, 2023, Art. no. 4827, doi: 10.3390/electronics12234827.
20. N. Sharma, V. Jain, and A. Mishra, “An analysis of convolutional neural networks for image classification,” *Procedia Computer Science*, vol. 132, pp. 377–384, 2018, doi: 10.1016/j.procs.2018.05.198.
21. M. Tzelepi and A. Tefas, “Deep convolutional learning for content based image retrieval,” *Neurocomputing*, vol. 275, pp. 2467–2478, January 2018, doi: 10.1016/j.neucom.2017.11.022.
22. J. Cheng, M. Sadiq, O. A. Kalugina, S. A. Nafees and Q. Umer, “Convolutional neural network based approval prediction of enhancement reports,” *IEEE Access*, vol. 9, pp. 122412–122424, 2021, doi: 10.1109/ACCESS.2021.3108624.
23. Y. H. Ali *et al.*, “Optimization system based on convolutional neural network and internet of medical things for early diagnosis of lung cancer,” *Bioengineering*, vol. 10, no. 3, 2023, Art. no. 320, doi: 10.3390/bioengineering10030320.
24. A. F. Al-Zubidi, A. K. Farhan, and S. M. Towfek, “Predicting DoS and DDoS attacks in network security scenarios using a hybrid deep learning model,” *Journal of Intelligent Systems*, vol. 33, no. 1, Art. no. 20230195, 2024, doi: 10.1515/jisys-2023-0195.
25. M. Manikandakumar and P. Karthikeyan, “Weed classification using particle swarm optimization and deep learning models,” *Computer Systems Science & Engineering*, vol. 44, no. 1, pp. 913–927, 2023, doi: 10.32604/csse.2023.025434.
26. M. Hong, B. Rim, H. Lee, H. Jang, J. Oh, and S. Choi, “Multi-class classification of lung diseases using CNN models,” *Applied Sciences*, vol. 11, no. 19, Art. no. 9289, 2021, doi: 10.3390/app11199289.