Journal of Contemporary Dental Sciences

Original Article

Predicting difficulty level of surgical removal of impacted mandibular third molar using deep learning approaches

Samir Ali Elborolosy[1]*, Walid S. Salem[2], Mohammed Omran Hamed[1], Arwa Saad Sayed[3], Bahaa El-din Helmy[3], Ahmed A. Elngar[3]

[1]Lecturer, Oral and Maxillofacial Surgery Department Faculty of Dentistry, Beni-Suef University, Beni-Suef, Egypt, [2]Associate Professor, Oral and Maxillofacial Radiology Department Faculty of Dentistry, Beni-Suef University, Beni-Suef City, Egypt, [3]Faculty of Computers and Artificial Intelligence, Beni-Suef University, Beni-Suef City, Egypt.

Corresponding Author:
Dr. Samir Ali Elborolosy,
E-mail: Dr.samir.elbrolose@dent.bsu.edu.eg

Access the journal online

Website:
https://jcds.qu.edu.sa/index.php/JCDS

e-ISSN: 1658-8207
p-ISSN: 1658-8193
PUBLISHER: Qassim University

ABSTRACT

Objective: This work presents three automated pre-trained models to predict the difficulty of extracting the mandibular third molar using a dataset of 2414 panoramic radiography images based on pre-processed (shifted and rotated) from the left and right mandibular third molar instances.

Methods: We employed four distinct architectural models, namely, VGG-16, VGG-19, MobileNetV2, and ResNet50 to identify the difficulty of removing a mandibular third molar. We categorized the dataset into four categories of complexity to help in categorization (normal, easy, medium, and difficult).

Results: VGG-16, VGG-19, MobileNetV2 and ResNet50 had prediction accuracies of 81%, 82%, 79% and 44%, respectively.

Conclusions: The proposed deep learning model using VGG-19 could be a good tool to predict the difficulty of extracting a mandibular third molar using a panoramic radiographic image.

Keywords: Deep learning, difficulty level, mandibular third molar, prediction accuracies, surgical removal

Introduction

Artificial intelligence is defined as a system's ability to accurately understand external input, learn from it, and apply what it has learned to achieve specific objectives and tasks through flexible adaptation.[1] The digital revolution in dentistry has significantly transformed the discipline by largely automating the traditional dental process. Digital advancements have smoothed and accelerated daily practice, as well as providing great ease of use in several sectors, resulting in significant time and cost savings. It has shown tremendous development in dentistry through the use of deep learning (DL) models by detecting patterns in huge amounts of data and acquiring important data to gain additional knowledge and solve dental problems, such as the detection of carious lesions, periodontal lesions, mandibular canals, and cysts. They are also used to classify the skeleton and determine the difficulty of removing the third molar.[2]

One of the most common procedures in oral surgery is the removal of the third molar. Several studies have evaluated the difficulties of this surgical technique, with most of them attempting to identify the primary risk factors.[3] The Pedersen Scale classifies third molars using the Pell and Gregory classification (position of the third molar in relation to the occlusal plane and mandibular ramus) as well as tooth angulation.[4] On the other hand, the modified parent scale assesses the procedure's difficulty using surgical technique parameters.[5] Even though these classifications are

routinely used to estimate the difficulty of third molar extraction, they both disregard some key pre-operative and intraoperative variables. The sensitivity of the Pedersen scale is compromised because it is based only on radiological data.

Accurately predicting a disease outcome is one of the most interesting and difficult jobs for clinicians.[6] Despite the widespread use of machine learning in medical research, attempts to employ it in disease diagnosis and prognosis are still relatively new, with the majority involving detection and classification.[7] Hence, medical imaging and computer-assisted surgical planning are important components of the pre-operative workup[8,9] because experimenting with different operating procedures in a virtual environment can reduce operation time and cost[10] while also promoting more consistent and optimized outcomes.

DL is also widely employed in biomedical imaging. Most DL methods rely on many data samples for training due to the need to optimize an immense number of weighting factors in convolutional neural networks (CNNs). The CNNs are designed to learn patterns from large datasets, without the need for a supervisor to label the data. The term "deep" refers to the number of (hidden) network layers to progressively extract information and features from the input data. The layers are interconnected via nodes or neurons. Each hidden layer uses the output of the previous layer as its input, thereby increasing the complexity and detail of what it is learning from layer to layer[11] Unfortunately, data collection in medical informatics is seen as a time-consuming and costly procedure. In the clinical setting, obtaining many training datasets is difficult, resulting in the overfitting of the model due to the constraints of small datasets.[12] As a result, a modified deep CNN has been studied to overcome the problem of limited datasets, which is based on transfer learning of unsupervised pre-training from a large number of datasets.[13]

Many patients have their mandibular third molars removed for a variety of reasons.[14,15] As a result, mandibular third molar extraction is a common procedure in oral and maxillofacial surgery. Symptoms appear 30–68% of the time following extraction of the third molar, depending on the impaction type of the third molar.[16] Third molars in the mandible grow in a variety of locations and directions, resulting in a variety of impaction patterns.[17] As a result, it is critical to evaluate the pattern of the impacted mandibular third molar before extraction to apply the proper surgical approach based on the impaction pattern.

Wisdom teeth are of great importance, but sometimes they have great damage, and in this case, the best solution is to remove them. Hence, in this study, we developed CNN to determine how difficult to remove a mandibular third molar. We used various transfer learning from the sample dataset on our panoramic radiology dataset with different classes (difficult, easy, medium, and normal), tested the system by comparing accuracy for different models such as VGG-16, VGG-19, MobilnetV2, and ResNet50.

Materials and Methods

Data preparation

Image data augmentation is a technique for artificially increasing the size of a training dataset by modifying photographs in the dataset. More data can help DL neural network models become more skilled, and augmentation approaches can help fit models generalize what they have learned to new images by creating modifications of the images. The Image Data Generator class in the Keras DL neural network toolkit allows you to fit models with image data, as shown in Figure 1.

Histogram equalization techniques for image enhancement show the images how the pixels in the image are distributed in terms of intensity. The histogram of photos that are too light or too dark.[18]

Histogram Equalization is widely used and developed, with multi-histogram Equalization being utilized to improve image contrast and brightness. The average image intensity of an image output produced by a dynamic equalization histogram is equal to the average image intensity of the input image. The histogram Equalization approach can be used not just in pictures but also in videos, resulting in a brilliant image output.[19] Improved image quality is a technique for achieving specific image conditions, as shown in Figure 2.

One of the most significant processes in grayscale picture data analysis is the segmentation algorithm. The threshold segmentation technique, among numerous grayscale picture segmentation algorithms, has the advantages of simplicity and efficiency of implementation and is thus commonly used.[20]

The OTSU algorithm searches for thresholds by exhausting all solutions in the gray space, so as the number of thresholds grows, the search dimension to be performed grows as well, increasing the complexity. Many unnecessary calculations are performed, the time grows exponentially, and the search efficiency decreases.[21] Furthermore, images acquired through various channels are subject to a variety of random disturbances and conditions, resulting in a large amount of noise in the acquired original images, causing the features of things in the acquired original images to change dramatically, and if such images are analyzed directly, the understanding of the images will be greatly skewed.[22] As a result, optimizing the OTSU algorithm to increase computing efficiency and efficacy has become a challenging and contentious subject. In this research, adaptive and fast methods are investigated for improving the OTSU algorithm's segmentation efficiency and optimizing the segmentation effect, as shown in Figure 3.

Pre-trained models in DL

A pre-trained model is a model created by someone else to solve a similar problem. Instead of building a model from scratch to solve a similar problem, you use the model trained on another problem as a starting point.

As the DL-based network evolves, its structure is deepening; while this helps the network to perform more complex feature pattern extraction, it may also introduce the problem of gradient disappearance or gradient explosion. "Gradient disappearance" and "gradient explosion" can lead to the following shortcomings: (1) Long training time but network convergence becomes very difficult or even non-convergent. (2) The network performance will gradually saturate and even begin to degrade, known as the degradation problem of deep networks. To solve such problems, He *et al.*[23] proposed the ResNet network, which makes it possible to obtain good performance and efficiency of the network even when the number of network layers is very deep (even

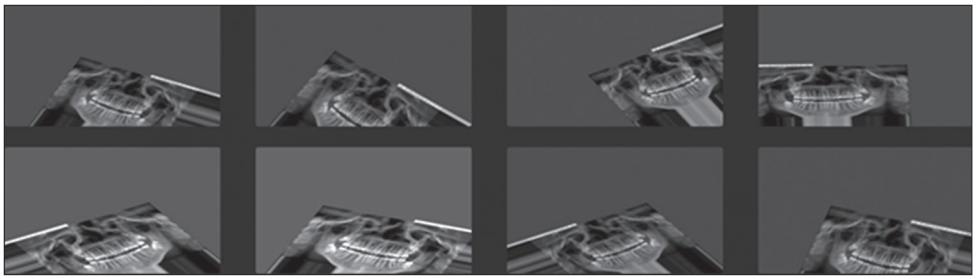

Figure 1: A training dataset

Figure 2: Histogram equalization

Figure 3: OTSU algorithm's segmentation

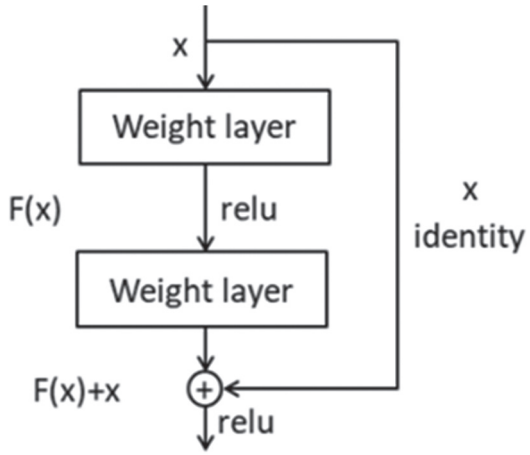

Figure 4: Pre-trained models

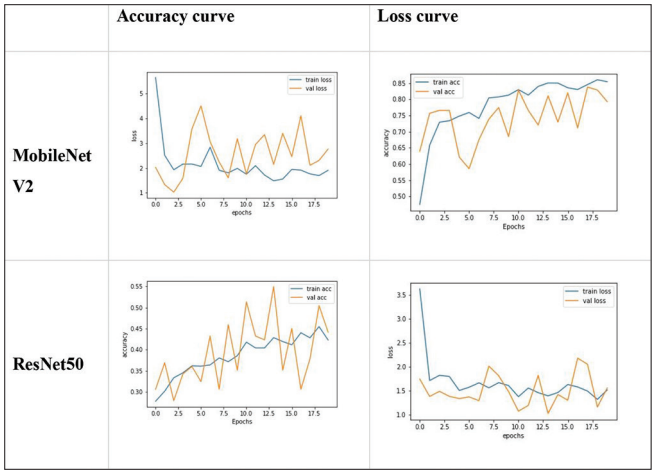

Figure 5: Accuracy and loss curves for the VGG-16 and VGG-19

over 1000 layers), as shown in Figure 4.

There is an identity mapping in the residual module of ResNet that causes the output of the network to change from F (x) to F (x) + x. The training error of a deep network is generally higher than that of a shallow network. However, adding multiple layers of constant mapping (y = x) to a shallow network turns it into a deep network, and such a deep network can get the same training error as a shallow network. This shows that the layers of constant mapping are better trained. For the residual network, when the residual is 0, the stacking layer only does constant mapping at this time, and according to the above conclusion, theoretically, the network performance will not degrade at least.

VGG16 is a 16-layer network used by the Visual Geometry Group at the University of Oxford to obtain state-of-the-art results in the ILSVRC-2014 competition. The main feature of this architecture was the increased depth of the network. In VGG16, 224 × 224 RGB images are passed through 5 blocks of convolutional layers where each block is composed of increasing numbers of 3 × 3 filters. The stride is fixed to 1 while the convolutional layer inputs are padded such that the spatial resolution is preserved after convolution (i.e., the padding is 1 pixel for 33 filters). The blocks are separated by max-pooling layers. Max-pooling has performed over 22 windows with stride 2. The 5 blocks of convolutional layers are followed by three fully connected (FC) layers. The final layer is a soft-max layer that outputs class probabilities Figure 5.

The VGG-19 neural network consists of 19 layers of deep neural network and has more weight. The size of the "VGG-19" network in terms of FC nodes is 574 MB. As the number of layers increases, the accuracy of DNN is improved. The VGG-19 model is comprised of 19 deep trainable layers performing convolution, which is FC with max-pooling and dropout layers. In this paper, the convolutional layer is trained to perform a customized classification role that involved a densely connected classifier and to regularize a dropout layer was used.

VGG-19 is so beneficial, and it simply uses 3 × 3 ConvNet arranged as above to extend the depth. To decrease the size, max-pooling layers are applied as a handler. Fully convolutional network layers are two in number and have 4096 neurons applied. VGG is trained based on individual lesions and for testing all types of lesions were considered to reduce the number of false positives. Convolution layers perform the convolution process over images at every pixel, allowing the outcome to pass through the subsequent layer Figure 5.

MobileNets is a model built primarily from depth-wise separable convolutions that were first announced in Sifre[24] and later utilized in Inception models[25] to reduce the volume of computation restrictions in the first few layers. Flattened networks[26] built a network comprised of fully factorized convolutions and showed the possibility of extremely factorized networks. In addition to using topological networks, Factorized Networks[27] uses a similar factorized convolution. The Xception network[28] was the first to show how to scale up depth-wise separable filters to outperform. Inception V3 networks. Squeeze net[29] is another small network that uses the bottleneck method to create a very small network. Arranged transform networks[30] and deep-fried ConvNets are two other reduced computation networks.

DL

DL is a variant of the traditional neural network that outperforms its predecessors significantly. Furthermore, DL uses both transformations and graph technology to construct multi-layer learning models. The latest DL algorithms have achieved excellent results in a range of applications, including audio and speech processing, image processing, visual data processing, and natural language processing (NLP), among others.[31]

Types of DL

Recursive neural networks (RVNN)

RvNN can predict outcomes in a hierarchical framework and classify them using compositional vectors.[32] RvNN development is primarily inspired by recursive auto-associative memory. The RvNN architecture was created to handle objects with randomly formed structures, such as graphs and trees. From a variable-size recursive-data structure, this method constructs a fixed-width distributed representation. A back-propagation through structure (BTS) learning mechanism is used to train the network.[33]

The BTS system uses the same general-back propagation method as the general-back propagation algorithm and can support a treelike structure. The network is trained to reproduce the input-layer pattern at the output layer through auto-association. In the domain of NLP, RvNN is extremely effective.[34]

Recurrent neural networks (RNNs)

RNNs are a well-known and widely used algorithm in the field of DL. RNN is mostly used in speech processing and natural language processing.[35] RNNs, unlike traditional networks, utilize sequential data in their networks. This property is essential to a variety of applications because the inherent structure in the data sequence provides valuable information. To discern the meaning of a given word in a sentence, for example, it is necessary to comprehend the context of the statement. As a result, the RNN can be thought of as a short-term memory unit, with x representing the input layer, y representing the output layer, and s representing the state (hidden) layer. Three different types of deep RNN techniques, namely, "Hidden-to-Hidden," "Hidden-to-Output," and "Input-to-Hidden."

CNNs

The CNN algorithm is the most well-known and widely used.[36] The fundamental advantage of CNN over its predecessors is that it automatically recognizes relevant elements without the need for human intervention.[37] Computer vision,[38] audio processing,[39] and face recognition[40] are just a few of the disciplines where CNNs have been used widely.

Like a traditional neural network, the structure of CNNs was inspired by neurons in human and animal brains. A complicated sequence of cells creates the visual cortex in a cat's brain, and this pattern is mimicked by CNN.[41] Three major advantages of CNN have been identified: equivalent representations, sparse interactions, and parameter sharing. Unlike traditional FC networks, CNN uses shared weights and local connections to fully exploit 2D input-data structures like image signals.

Each layer's input x is structured in three dimensions in a CNN model: height, width, and depth, or m*m*r, where the height (m) equals the width. The channel number is another name for the depth. The depth (r) of an RGB image, for example, is three. Each convolutional layer has several kernels (filters) that are designated by k and have three dimensions (n*n*q), comparable to the input picture; however, n must be smaller than m, and q must be equal to or smaller than r. Furthermore, the kernels serve as the foundation for local connections, which use comparable parameters (bias [bk] and weight [wk]) to generate k feature maps. (m-n-1) with the size hk. The dot product between the input and the weights is calculated by the convolution layer.

$$hk = f(Wk * x + bk) \quad (1)$$

Next, each feature map in the sub-sampling layers is down-sampled. This results in a decrease in network parameters, which speeds up the training process and allows for the resolution of the overfitting problem. The pooling function adjacent area of size p*p, where p is the kernel size, for all feature maps. Finally, as in a conventional neural network, the FC layers receive the mid-and low-level features and generate the high-level abstraction, which represents the last-stage layers. The final layer is used to provide classification scores. Every score represents the probability of a specific class for a given case.

CNN layers

Convolutional layer

It is made up of a set of convolutional filters (so-called kernels). The output feature map is generated by convolving the input image with these filters, which are expressed as N-dimensional metrics. Kernel definition: A grid of discrete numbers or values describes the kernel, the CNN input format is presented first, followed by the convolutional operation. The vector format is the classic neural network's input, while the CNN's input is a multi-channeled picture. The gray-scale image format, for example, is single-channel, but the RGB image format is three-channeled. Consider a 4*4 gray-scale image with a 2*2 random weight-initialized kernel to better comprehend the convolutional operation.

Pooling layer

The pooling layer's primary function is to subsample the feature maps. The convolutional operations are used to create these maps. In other words, this method reduces the size of huge feature maps to make smaller feature maps. At the same time, it keeps most of the dominating information (or characteristics) throughout the pooling stage. Before the pooling process, both the stride and the kernel are size assigned in the same way as the convolutional operation is. In different pooling layers, many types of pooling algorithms are accessible. Tree pooling, gated pooling, average pooling, min pooling, max pooling, global average pooling (GAP), and global max-pooling are some of these strategies. The most familiar and frequently utilized pooling methods are the max, min, and GAP pooling.

Activation function

It makes the decision as to whether or not to fire a neuron with reference to a particular input by creating the corresponding output. It must also be able to distinguish, which is a crucial characteristic because it allows the network to be trained using error back-propagation. In CNNs and other deep neural networks, the following types of activation functions are most typically utilized.

Sigmoid

This activation function takes real numbers as input and outputs only values between zero and one. The S-shaped sigmoid function curve can be quantitatively represented.

$$f(x)\ \text{sigm} = 1/(1+e-x)$$

Tanh

It's similar to the sigmoid function in that it takes real values as input, but the output can only be between −1 and 1.

$$f(x)\tanh = (ex - e-x)/(ex + e-x)$$

ReLU

The most commonly used function in the CNN context. It converts the whole values of the input to positive numbers. Lower computational load is the main benefit of ReLU over the others.

$$f(x)\text{ReLU} = \max(0,x)$$

FC layer

This layer is at the bottom of any CNN architecture. The so-called FC technique connects each neuron in this layer to all neurons in the previous layer. As a CNN classifier, it is used. As a feed-forward ANN, it uses the same basic mechanism as a traditional multiple-layer perceptron neural network. The FC layer gets its input from the previous pooling or convolutional layer. This input takes the shape of a vector, which is formed by flattening the feature maps.

Loss functions

The projected error created across the training samples in the CNN model is calculated using loss functions in the output layer. The disparity between the actual and expected output is revealed by this inaccuracy. The CNN learning process will then be used to optimize it. The loss function, on the other hand, uses two parameters to determine the error. The first parameter is the CNN estimated output (also known as the prediction). The second parameter is the actual output (sometimes known as the label). Several types of loss functions are employed in various problem types.

Cross-entropy or SoftMax Loss function

This function is frequently used to evaluate the CNN model's performance. The log loss function is another name for it. The likelihood is the output of the probability $p \in \{0,1\}$. In addition, multi-class classification issues, it is commonly used to replace the square error loss function. It uses SoftMax activations in the output layer to generate the output within a probability distribution.

$$pi = eak / \sum NK = 1\ eak$$

Here, eai represents the non-normalized output from the preceding layer, while N represents the number of neurons in the output layer.

$$H(p,y) = -\sum yi \log(pi)$$

Where $i \in [1, N]$.

Euclidean loss function

This function is widely used in regression problems. In addition, it is also the so-called mean square error.

$$H(p,y) = 1/2N * \sum(pi - yi)^2$$

Hinge loss function

This function is commonly employed in problems related to binary classification.

$$H(p, y) = \sum \max(0, m - [2yi - 1]pi)$$

The margin m is commonly set to 1. Moreover, the predicted output is denoted as pi, while the desired output is denoted as yi.

Results

In this study, four different pre-trained architectural models, namely, VGG-16, VGG-19, MobileNetV2, and ResNet50 are employed to identify the difficulty of removing a mandibular third molar. Four classes of difficulty are determined (easy, difficult, medium, and normal). This section shows and discusses the results obtained by each model using several metrics such as accuracy, precision, recall, sensitivity, and specificity. Accuracy could be defined as the number of correctly predicted data out of all the data as shown in Eq. (1). Precision defines the number of positive class predictions that belong to the positive class as shown in Eq. (2). Recall denotes the number of positive class predictions made out of all positive examples in the dataset as shown in Eq. (3). The sensitivity is the true positive rate that could be formulated as shown in Eq. (4). Moreover, specificity is the true negative rate and can be defined, as shown in Eq. (5).

Here, TP, TN, FP, and FN refer to the true positive, true negative, false positive, and false-negative cases.

$$\text{Accurancy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3)$$

$$\text{Sensitvity} = \frac{TP}{TP + FN} \quad (4)$$

$$\text{Specificity} = \frac{TN}{TN + FP} \quad (5)$$

Discussion

VGG-16 and VGG-19 results

Table 1 shows the results obtained from VGG-16 and VGG-19 in terms of accuracy, precision, recall, sensitivity, and specificity measures. It is noticeable that VGG-16 can efficiently classify the cases of medium class. Moreover, the overall training accuracy is also better than VGG-19. On the other hand, the VGG-19 produces good results in difficult, easy, and normal cases. Furthermore, VGG-19 gains a promising accuracy, especially the testing accuracy. Hence, we can say that

Table 1: Accuracy, Precision, Recall, Sensitivity, Specificity results for VGG-16 and VGG-19.

	VGG-16		VGG-19	
	Precision	Recall	Precision	Recall
Difficult	0.17	0.11	0.26	0.26
Easy	0.13	0.13	0.32	0.37
Medium	0.26	0.37	0.19	0.20
Normal	0.2	0.17	0.21	0.17
Train accuracy	0.89		0.84	
Test accuracy	0.81		0.82	
Sensitivity	0.27		0.46	
Specificity	0.33		0.57	
Time	1h 17m		1h 39m	

Table 2: Show the Accuracy, Precision, Recall, Sensitivity, Specificity results for ResNet50 and MobileNetV2

	MobileNetV2		ResNet50	
	Precision	Recall	Precision	Recall
Difficult	0.19	0.26	0.23	0.48
Easy	0.29	0.33	0	0
Medium	0.13	0.07	0	0
Normal	0.23	0.25	0.13	0.29
Train accuracy	0.85		0.40	
Test accuracy	0.79		0.44	
Sensitivity	0.5		0.92	
Specificity	0.52		0	
Time	14m		40m	

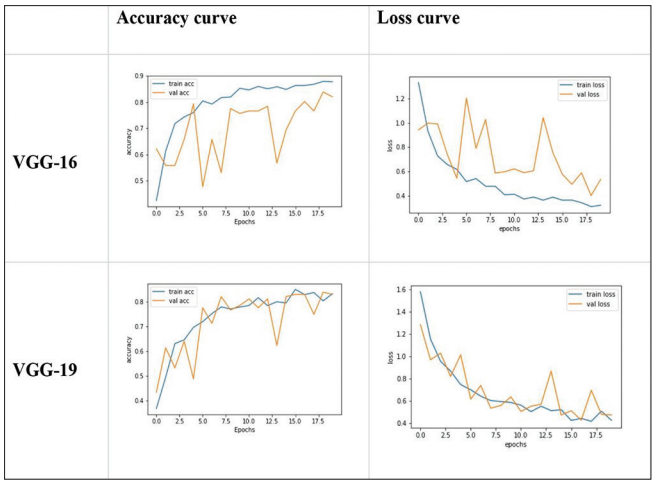

Figure 6: Accuracy and Loss curves for the MobileNet, and ResNet models

VGG-19 is better than VGG-16 in terms of precision, recall, accuracy, sensitivity, and specificity. Although, the VGG-19 takes a long run time than the VGG-16. Table 2 provide the accuracy and loss curves for each pre-trained model and we can notice that VGG-19 has a good/smooth accuracy and loss curves compared to the VGG-16 model.

MobileNetV2 and ResNet50 results

Table 2 shows the results obtained from MobileNetV2 and ResNet50 in terms of accuracy, precision, recall, sensitivity, and specificity measures. It is noticeable that MobileNetV2 can efficiently classify the cases of easy, medium, and normal classes. Moreover, the overall training accuracy is also better than ResNet50. The time required to completely run MobileNetV2 is less than the ResNet50. Hence, we can say that MobileNetV2 produces good classification results compared to the ResNet50. Figure 6 shows the accuracy and loss curves for both MobileNetV2 and ResNet50 models. Moreover, MobileNetV2 the accuracy and loss curves in the case of using MobileNetV2 are good compared to the ResNet50 model.

Conclusions

This study presented and validated DL tools for fast, accurate, and consistent automated measurement of removing a mandibular third molar on dental panoramic radiographs. In this work, the VGG-19 and MobileNetV2 produce promising results. Moreover, the MobileNetV2 model is a very fast-running model compared to other models. Hence, VGG-19 and MobileNetV2 models can effectively predict the difficulty of extracting the mandibular third molar.

Ethics Approval and Consent to Participate

The study was approved by the faculty of dentistry Beni-Suef University Research Ethics Committee (FDBSUREC) under the Approval number: #FDBSUREC/09072020/AS.

Author Contributions

All authors' individual contributions, using the relevant CRediT roles: Conceptualization; Dr Samir Ali Elborolosy, Data curation; Walid S Salem, Formal analysis; Dr. Mohammed Omran Hamed, Investigation: Ahmed A. Elngar, Methodology; Bahaa El-din Helmy, Project administration; Ahmed A. Elngar, Resources; Dr Samir Ali Elborolosy, Software; Arwa Saad Sayed, Supervision; Walid S Salem, Validation; Dr. Mohammed Omran Hamed, Visualization; Bahaa El-din Helmy, Roles/Writing – original draft: Arwa Saad Sayed, Writing – review and editing. Ahmed A. Elngar

Funding

This research received no external funding.

Data Availability Statement

All data generated or analyzed as part of this study are included in this published article.

Acknowledgments

This study was supported by the Beni-Suef University Faculty of Dentistry.

Conflicts of Interest

Neither the author nor any of the co-authors have any potential conflict of interest related to the publication of this paper. The authors state that there are no financial and personal relationships with other people or organizations that could inappropriately influence their work.

References

1. Kaplan A, Haenlein M. Siri, Siri, in my hand: Who's the fairest in the land? On the interpretations, illustrations, and implications of artificial intelligence. Bus Horiz 2019;62:15-25.
2. Jaskari J, Sahlsten J, Järnstedt J, Mehtonen H, Karhu K, Sundqvist O, *et al*. Deep learning method for mandibular canal segmentation in dental cone beam computed tomography volumes. Sci Rep 2020;10:5842.

3. Akadiri O, Obiechina A. Assessment of difficulty in third molar surgery--a systematic review. J Oral Maxillofac Surg 2009;67:771-4.
4. Koerner K. The removal of impacted third molars. Principles and procedures. Dent Clin North Am 1994;38:255-78.
5. Diniz-Freitas M, Lago-Méndez L, Gude-Sampedro F, Somoza-Martin JM, Gándara-Rey JM, García-García A. Pederson scale fails to predict how difficult it will be to extract lower third molars. Br J Oral Maxillofac Surg 2007;45:23-6.
6. Kourou K, Exarchos TP, Exarchos KP, Karamouzis MV, Fotiadis DI. Machine learning applications in cancer prognosis and prediction. Comput Struct Biotechnol J 2015;13:8-17.
7. Cruz JA, Wishart DS. Applications of machine learning in cancer prediction and prognosis. Cancer Inform 2007;2:59-77.
8. Meulstee J, Liebregts J, Xi T, Vos F, de Koning M, Bergé S, *et al*. A new 3D approach to evaluate facial profile changes following BSSO. J Craniomaxillofac Surg2015;43:1994-9.
9. Steinbacher DM. Three-dimensional analysis and surgical planning in craniomaxillofacial surgery. J Oral Maxillofac Surg 2015;73:S40-56.
10. Steinhuber T, Brunold S, Gärtner C, Offermanns V, Ulmer H, Ploder O. Is virtual surgical planning in orthognathic surgery faster than conventional planning? A time and workflow analysis of an office-based workflow for single- and double-jaw surgery. J Oral Maxillofac Surg 2018;76:397-407.
11. Pesapane F, Codari M, Sardanelli F. Artificial intelligence in medical imaging: Threat or opportunity? Radiologists again at the forefront of innovation in medicine. Eur Radiol Exp 2018;2:35.
12. Min S, Lee B, Yoon S. Deep learning in bioinformatics. Brief Bioinform 2017;18:851-69.
13. Shin HC, Roth HR, Gao M, Lu L, Xu Z, Nogues I, *et al*. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. IEEE Trans Med Imaging 2016;35:1285-98.
14. Kruger E, Thomson WM, Konthasinghe P. Third molar outcomes from age 18 to 26: Findings from a population-based New Zealand longitudinal study. Oral Surg Oral Med Oral Pathol Oral Radiol Endod 2001;92:150-5.
15. Alfadil L, Almajed E. Prevalence of impacted third molars and the reason for extraction in Saudi Arabia. Saudi Dent J 2020;32:262-8.
16. Yilmaz S, Adisen MZ, Misirlioglu M, Yorubulut S. Assessment of third molar impaction pattern and associated clinical symptoms in a central anatolian turkish population. Med Princ Pract 2016;25:169-75.
17. Jaroń A, Trybek G. The pattern of mandibular third molar impaction and assessment of surgery difficulty: A retrospective study of radiographs in East baltic population. Int J Environ Res Public Health 2021;18:6016.
18. Tan K, Oakley JP. Enhancement of Color Images in Poor Visibility Conditions. In: Proceedings 2000 International Conference on Image Processing. Vol. 2. IEEE; 2000.
19. Chen SD, Ramli AR. Minimum mean brightness error bi-histogram equalization in contrast enhancement. IEEE Trans Consum Electron 2003;49:1310-9.
20. Ding W, Zhao Y, Zhang R. An adaptive multi-threshold segmentation algorithm for complex images under unstable imaging environment. Int J Comput Appl Technol 2019;61:265-72.
21. Shao D, Xu C, Xiang Y, Gui P, Zhu X, Zhang C, *et al*. Ultrasound image segmentation with multilevel threshold based on differential search algorithm. IET Image Process 2019;13:998-1005.
22. Sun S, Song H, He D, Long Y. An adaptive segmentation method combining MSRCR and mean shift algorithm with K-means correction of green apples in natural environment. Inf Process Agric 2019;6:200-15.
23. He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, NV, USA; 2016.
24. Sifre L. Rigid Motion Scattering for Image Classification. Ph.D thesis; 2014.
25. Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv [Preprint arXiv:1502.03167]. 2015.
26. Jin J, Dundar A, Culurciello E. Flattened convolutional neural networks for feedforward acceleration. arXiv [Preprint arXiv:1412.5474]. 2014.
27. Wang M, Liu B, Foroosh H. Factorized convolutional neural networks. arXiv [Preprint arXiv:1608.04337]. 2016.
28. Chollet F. Xception: Deep learning with depthwise separable convolutions. arXiv [Preprint arXiv:1610.02357v2]. 2016.
29. Iandola FN, Moskewicz MW, Ashraf K, Han S, Dally WJ, Keutzer K. Squeezenet: Alexnet-level accuracy with 50x fewer parameters andi 1MB model size. arXiv [Preprint arXiv:1602.07360]. 2016.
30. Sindhwani V, Sainath T, Kumar S. Structured transforms for small-footprint deep learning. In Advances in Neural Information Processing Systems. United States: MIT Press; 2015. p. 3088-96.
31. Koppe G, Meyer-Lindenberg A, Durstewitz D. Deep learning for small and big data in psychiatry. Neuropsychopharmacology 2021;46:176-90.
32. Socher R, Perelygin A, Wu J, Chuang J, Manning CD, Ng AY, Potts C. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. Washington, DC: Association for Computational Linguistics; 2013.
33. Goller C, Kuchler A. Learning Task-dependent Distributed Representations by Backpropagation through Structure. In: Proceedings of International Conference on Neural Networks (ICNN'96). Vol. 1. IEEE; 1996.
34. Socher R, Lin CCY, Ng AY, Manning CD. Parsing natural scenes and natural language with recursive neural networks. In: Proceedings of the 28th International Conference on Machine Learning, (ICML); 2011.
35. Jiang Y, Kim H, Asnani H, Kannan S, Oh S, Viswanath P. Learn codes: Inventing low-latency codes via recurrent neural networks. IEEE J Sel Areas Inf Theory 2020;1:207-16.
36. Zhou DX. Theory of deep convolutional neural networks: Downsampling. Neural Netw 2020;124:319-27.
37. Gu J, Wang Z, Kuen J, Ma L, Shahroudy A, Shuai B, *et al*. Recent advances in convolutional neural networks. Pattern Recogn 2018;77:354-77.
38. Fang W, Love PE, Luo H, Ding L. Computer vision for behaviour-based safety in construction: A review and future directions. Adv Eng Inform 2020;43:100980.
39. Palaz D, Magimai-Doss M, Collobert R. End-to-end acoustic modeling using convolutional neural networks for hmm-based automatic speech recognition. Speech Commun 2019;108:15-32.
40. Li HC, Deng ZY, Chiang HH. Lightweight and resource-constrained learning network for face recognition with performance optimization. Sensors (Basel) 2020;20:6114.
41. Hubel DH, Wiesel TN. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. J Physiol 1962;160:106-54.