



A Comparative Analysis of Deep Learning-Based Approaches for Classifying Dental Implants Decision Support System

Mohammed A. H. Lubbad^{1,5} · Ikbaleblebicioglu Kurtulus² · Dervis Karaboga^{1,5} · Kerem Kilic² · Alper Basturk^{1,5} · Bahriye Akay^{1,5} · Ozkan Ufuk Nalbantoglu^{1,5} · Ozden Melis Durmaz Yilmaz² · Mustafa Ayata² · Serkan Yilmaz³ · Ishak Pacal^{4,5}

Received: 20 January 2024 / Revised: 28 February 2024 / Accepted: 12 March 2024
© The Author(s) under exclusive licence to Society for Imaging Informatics in Medicine 2024

Abstract

This study aims to provide an effective solution for the autonomous identification of dental implant brands through a deep learning-based computer diagnostic system. It also seeks to ascertain the system's potential in clinical practices and to offer a strategic framework for improving diagnosis and treatment processes in implantology. This study employed a total of 28 different deep learning models, including 18 convolutional neural network (CNN) models (VGG, ResNet, DenseNet, EfficientNet, RegNet, ConvNeXt) and 10 vision transformer models (Swin and Vision Transformer). The dataset comprises 1258 panoramic radiographs from patients who received implant treatments at Erciyes University Faculty of Dentistry between 2012 and 2023. It is utilized for the training and evaluation process of deep learning models and consists of prototypes from six different implant systems provided by six manufacturers. The deep learning-based dental implant system provided high classification accuracy for different dental implant brands using deep learning models. Furthermore, among all the architectures evaluated, the small model of the ConvNeXt architecture achieved an impressive accuracy rate of 94.2%, demonstrating a high level of classification success. This study emphasizes the effectiveness of deep learning-based systems in achieving high classification accuracy in dental implant types. These findings pave the way for integrating advanced deep learning tools into clinical practice, promising significant improvements in patient care and treatment outcomes.

Keywords Dental implants · Periodontics · Prostheses and implants · Artificial intelligence · Deep learning

Introduction

Dental implants have become a standard and widespread solution for addressing tooth loss, with annual installations exceed 800,000 in the USA and 1.8 million in Europe, respectively [1]. The success rates of dental implants are

high; however, factors such as patient health, dentist technique, and implant suitability contribute to potential failures [2]. As the aging population increases, the demand for dental implants is expected to rise, emphasizing the need for careful selection of appropriate implants from manufacturers [3]. This is especially crucial due to the complexity and duration of the implant procedure. In clinical practice, complications related to dental implants, including biological and mechanical aspects, have been reported since their introduction to the market [4]. Technical complication rates vary, and it becomes imperative to possess detailed information, such as the implant manufacturer, system classification, diameter, and abutment type, to address challenges effectively [5].

Deep learning algorithms have emerged as powerful tools in addressing challenges within medical image processing, overcoming issues like low accuracy and manual feature extraction often associated with classical machine learning [6]. Widely applied in healthcare, defense, and

✉ Mohammed A. H. Lubbad
engmlubbad@gmail.com

¹ Department of Computer Engineering, Engineering Faculty, Erciyes University, 38039 Kayseri, Turkey

² Department of Prosthodontics, Dentistry Faculty, Erciyes University, Kayseri, Turkey

³ Department of Dentomaxillofacial Radiology, Dentistry Faculty, Erciyes University, Kayseri, Turkey

⁴ Department of Computer Engineering, Engineering Faculty, Iğdir University, Iğdir, Turkey

⁵ Artificial Intelligence and Big Data Application and Research Center, Erciyes University, Kayseri, Turkey

agriculture sectors, deep learning, especially through convolutional neural networks (CNNs) and vision transformers, facilitates swift and accurate diagnoses in fields like medical image processing [7, 8]. CNNs are specialized deep learning models designed specifically for image-processing tasks. They excel in tasks such as image classification and object detection, leveraging convolutional layers to detect and learn hierarchical patterns in images [9]. Conversely, vision transformers, a newer category of models, utilize a transformer architecture initially designed for natural language processing [10]. Vision transformers have demonstrated remarkable effectiveness in image classification by transforming input images into sequences of tokens, allowing the model to identify global dependencies and relationships within the dataset [11, 12]. In dental applications, these techniques prove invaluable, aiding in implant classification and the diagnosis of dental diseases [7].

To access implant components and associated structures, specific knowledge about the implant's brand is required, as each brand possesses unique screw and connection structures [13]. In dental applications, integrating image processing and deep learning techniques is crucial, assisting in tasks such as implant classification, dental disease diagnosis, and treatment planning. Deep learning technologies like CNNs and vision transformers promise to improve performance and generalization in medical imaging, including dental applications [14, 15].

However, a notable research gap exists regarding the clinical efficiency of deep learning algorithms in classifying dental implant systems [16]. This study aims to fill this gap by conducting experiments using various deep learning models, focusing on their clinical performance in implant dentistry. The proposed decision system classifies implant brands, offering advantages in terms of time and effort for both patients and dentists. Additionally, we expand the dental implant dataset to six classes, enhancing the system's reliability and applicability in different dental clinics. The contributions of our study are notably significant in the realm of dental implantology and the application of deep learning technologies. They can be succinctly outlined as follows:

- **Creation of a novel dataset:** We curated a unique dataset of dental implant images, meticulously collected by specialized dentists at the Faculty of Dentistry, Erciyes University in Turkey. This dataset is foundational for the research and application of deep learning in dental implant classification.
- **Dataset annotation and labeling:** The dataset was thoroughly annotated and labeled into six distinct classes, setting the stage for precise and meaningful classification efforts. This detailed categorization is crucial for the training and evaluation of deep learning models.

- **Evaluation of cutting-edge deep learning architectures:** Our study involved the comprehensive evaluation of recent deep learning architectures, including ConvNext, RegNet, DenseNet, VGG, and vision-based transformers. We leveraged transfer learning and data augmentation techniques to maximize the potential of these models for our specific application.
- **Fine-tuning transformers:** We fine-tuned selected transformer models specifically to address the classification challenge, significantly improving accuracy. This approach demonstrates the adaptability and effectiveness of transformer models in domain-specific applications.
- **Advancing the application of deep learning in implantology:** The primary focus of our article is to highlight and explore the application of deep learning technology in identifying dental implant brands from images. This addresses a critical and relatively unexplored aspect of dental implantology, showcasing the potential of AI to revolutionize this field.

Together, these contributions not only advance the scientific understanding and technological capabilities in the classification of dental implants but also open up new pathways for the practical application of deep learning in dental medicine, improving diagnostic accuracy and patient outcomes. The remainder of the paper is structured as follows: The “[Materials and Methods](#)” section outlines the materials and methods employed, “[Experiments and Discussion](#)” section presents the experiments and subsequent discussions, and “[Conclusion](#)” section concludes with final remarks and findings.

Materials and Methods

Deep Learning in Healthcare

As a machine-learning approach, deep learning encompasses using artificial neural networks to analyze extensive datasets [17]. Deep learning can analyze medical images, such as X-rays and MRIs, within the healthcare domain to detect diseases and medical conditions [18, 19]. Additionally, it can leverage medical records and other pertinent data to identify patterns and predict patient outcomes. Integrating deep learning in healthcare can significantly enhance the precision and effectiveness of medical diagnoses and treatments, ultimately leading to improved patient outcomes. Deep learning architectures are specifically designed artificial neural networks that are capable of learning hierarchical representations of data [20, 21]. These architectures typically consist of multiple layers of interconnected nodes, with each layer responsible for extracting distinct levels of abstraction from the data. Among the most widely used deep learning

architectures are CNNs, recurrent neural networks, and feed-forward neural networks. These architectures find application in various domains, including image classification, natural language processing, and speech recognition [22]. Deep learning architectures can be classified into supervised architectures and unsupervised architectures. Supervised architectures, including CNNs and recurrent neural networks (RNNs), are trained with labeled data and commonly used for tasks like image classification and sequence modeling [23]. On the other hand, unsupervised architectures, including Autoencoders, Restricted Boltzmann Machines (RBMs), and Deep Boltzmann Machines (DBMs), do not require labeled data for training and are often used for tasks like data representation learning and generative modeling [24]. However, it is worth noting that certain architectures, like generative adversarial networks (GANs), exhibit characteristics of both supervised and unsupervised approaches depending on their training and utilization methods. GANs are known for their ability to generate new data samples that resemble the training data distribution, making them valuable for tasks such as image synthesis and data augmentation [25].

In summary, deep learning offers immense potential in healthcare by enabling the analysis of medical images and data, leading to improved medical diagnoses and treatment outcomes. The diversity of deep learning architectures provides flexibility in addressing various tasks across different domains, empowering researchers and practitioners in their quest for innovative solutions.

CNNs

CNNs have become the most used deep learning techniques, providing the highest throughput among other artificial intelligence algorithms [26]. Researchers have collaborated by proposing new approaches to increase the efficiency of CNNs [27]. Different typical and recent deep convolutional neural network (DCNN) architectures, which

are now utilized as building blocks in various classification, segmentation, and detection designs, will be briefly discussed in this section. The earliest CNN model, the LeNet architecture, was described in research by Yann LeCun [28]. CNNs gained prominence in 2012 when the AlexNet design [29] won the ImageNet competition, offering a more robust structure with additional feature filters compared to LeNet. The most popular forms used as a backbone in CNN structures for object classification and detection of VGG design are VGG16 and VGG19. Also, with a 6.66% error rate and 22 layers, GoogLeNet [14] won the 2014 ILSVRC. This structure decreased the computational cost and the overfitting risk using the Inception modules [30]. In contrast, the architecture of ResNet [31] won the 2015 ILSVRC, which consists of 152 layers, with a 3.6% error rate. Later, several additional designs were introduced, including DenseNet [15], EfficientNet [32], and MobileNets [33]. VGG, Inception, and ResNet are the most often used architectures for disease analysis and image classification. In addition, while CNN structures were first employed for classification, they were widely adopted because of their effectiveness in object identification and segmentation applications [34].

This section provides a more comprehensive overview of the CNN architecture than other deep learning systems. The discussion encompasses a detailed examination of the construction of CNN architecture, including the description of its various topologies. Furthermore, notable CNN techniques employed in object recognition and image classification are elaborated upon, with specific emphasis on their application in the study of dental implants and the processing of medical images. An illustrative example of CNN architecture is depicted in Fig. 1 [35].

CNN Architectures

As the filter traverses the image from the previous layer or input image, the pixel values are multiplied, and the resulting values are summed and stored in the corresponding pixel

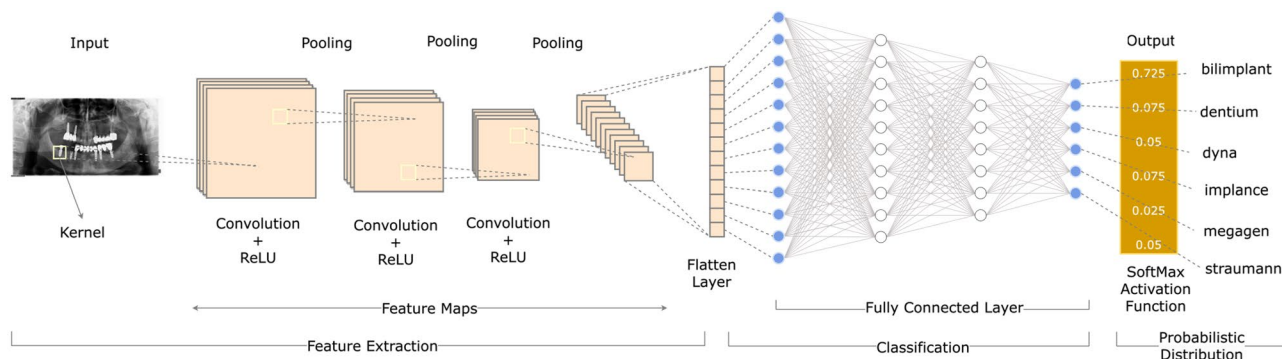


Fig. 1 A typical CNN architecture [35]

region of the output data [36]. This iterative process uses identical filters until the entire image has been analyzed [37]. The outcome is a feature map or matrix that encapsulates information about the distinct features identified by each filter. Mathematically, this convolution (S) process can be represented by Eq. 1 [38].

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(i + m, j + n).K(m, n) \quad (1)$$

where I represents the input image, K is the kernel, and (i, j) , m , and n are coordinates pertaining to the output feature map and the kernel, respectively. Through this process, as the kernel moves across the input image, it calculates the cumulative element-wise multiplication of the kernel with the corresponding segment of the image, thereby generating the output feature map. Consequently, this mechanism allows CNNs to discern and learn various filters that identify essential features of the image for a range of applications [39]. A convolutional layer incorporates multiple feature matrices, which are applied to either the input layer's image or the output of preceding layers [40]. As each cycle progresses, the characteristics within each column and row of the feature matrix are learned, utilizing the propagated gradient value for updating the column and row feature matrices [41]. By employing the same feature map and filters, weight

sharing occurs within the convolutional layer, subsequently reducing the network parameters' quantity [42]. The designers determine the filter number, width, padding, and stride parameters [28]. Conventionally, odd numbers serve as the dimensions for filters during the convolution process, facilitating balanced and centered padding. The stride parameter dictates the extent of the kernel's shift across the image. Padding ensures that the output size matches the input size by evenly adding zeros [43].

As illustrated in Fig. 2, an example of an output image generated from a convolution process employs a stride of 1, an input image of dimensions 4×4 , and a convolution core of dimensions 3×3 . Multiple kernels are utilized to identify varying characteristics, with repetition occurring to generate the required feature maps. Following convolution, Eq. 2 determines the output size [44].

output matrix dimension

$$= \frac{(\text{inputsize} + 2 * \text{padding} - \text{kernel size})}{\text{stride}} + 1 \quad (2)$$

For instance, in cases where the filter size is 3×3 and the input picture measures 64×64 , with a step shift of one, the resulting output matrix will have dimensions of 64×64 . If no padding is applied, the output matrix size reduces to 62×62 .

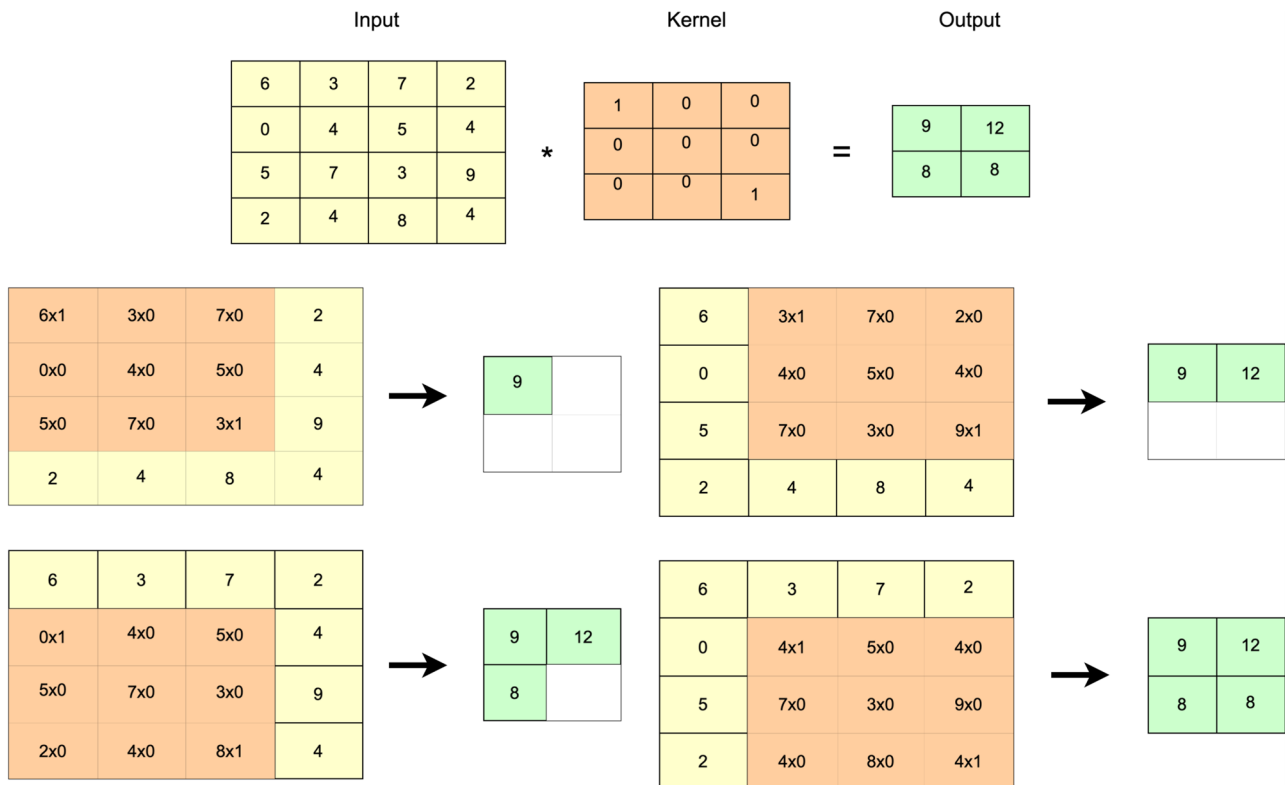


Fig. 2 A Convolution matrix with a kernel (3×3), an input picture (4×4)

Following the preceding layers, nonlinear activation functions, commonly referred to as nonlinearity layer functions, are employed to augment the network's nonlinearity. The incorporation of these activation functions has a profound impact on the performance of CNNs [30].

By promoting the independence of neurons in subsequent layers without excessive input values, these functions contribute to the generation of feature maps, thereby enhancing the overall stability of the network. Figure 3 depicts the most frequently utilized nonlinear activation functions, including Leaky ReLU, Softmax, ReLU, Sigmoid, Tanh, GELU, SiLU, Mish, and Swish. Leaky ReLU is a popular and effective method for tackling the dying ReLU problem, mathematically described in Eq. 3.

$$f_{\text{LeakyReLU}}(x) = \begin{cases} 0.01x & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases} \quad (3)$$

By introducing a slight slope in the negative range, Leaky ReLU functions as an improved version of ReLU, thereby mitigating the problem. The softmax function produces a vector of values that sum up to 1.0, allowing their interpretation as probabilities representing class membership. The sigmoid function yields outputs ranging from 0 to 1, making it particularly suitable for models requiring probability estimation. Similarly, the Tanh function ranges from -1 to 1 , where negative values indicate negativity, and zero values indicate neutrality. However, unlike sigmoid, the Tanh function exhibits symmetry. Nevertheless, the rectified linear unit (ReLU) has emerged as the most prevalent nonlinear function in CNNs. ReLU limits output to a range of $(0, x)$, mapping positive inputs directly and negative inputs to zero, without modifying the output. Due to its computational efficiency compared to Tanh and sigmoid functions and its ability to alleviate gradient vanishing issues, ReLU is widely

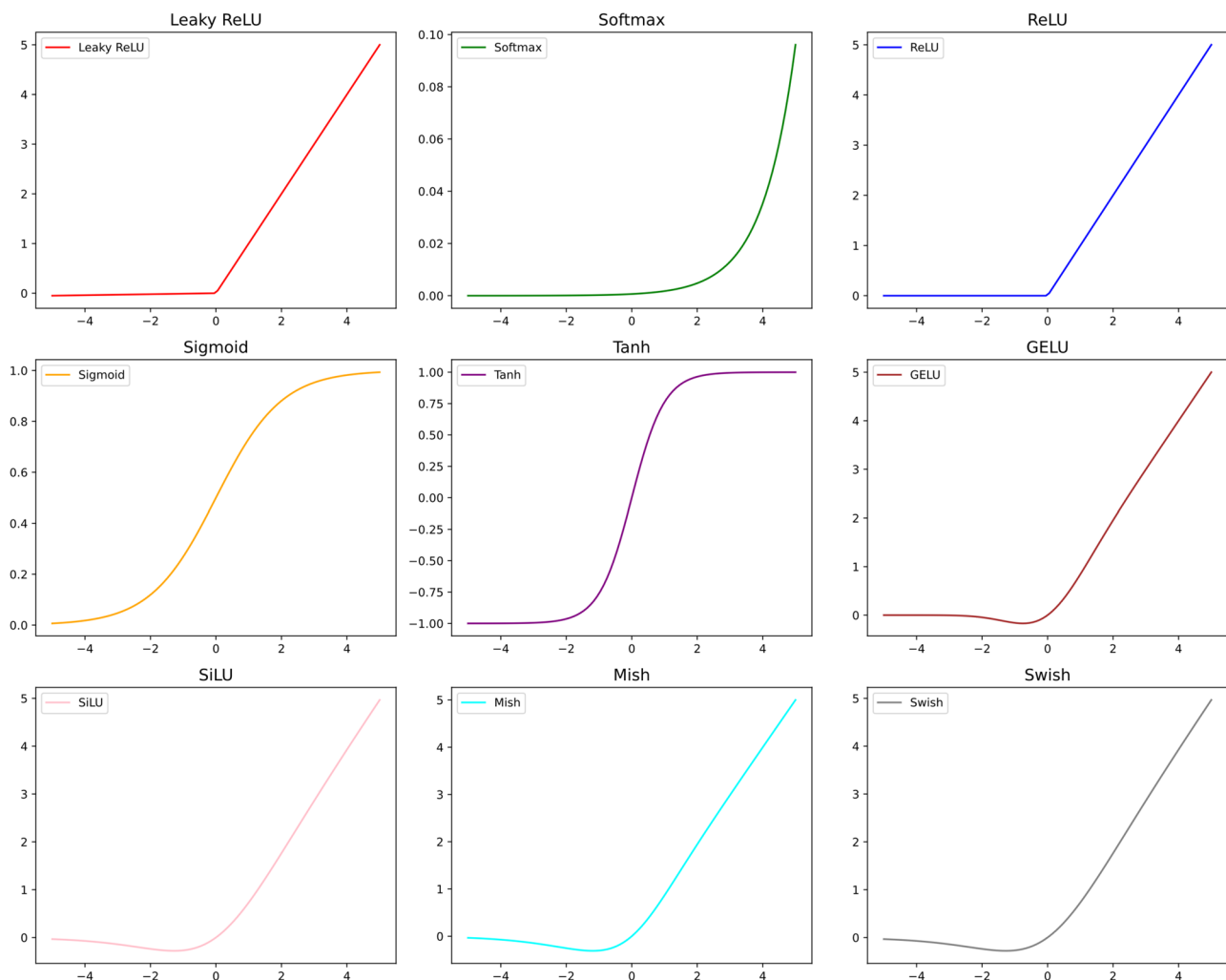


Fig. 3 Commonly used activation functions

employed in CNNs. These three well-known functions are defined mathematically in Eqs. 4, 5, and 6.

$$f_{\text{ReLU}}(x) = \max(0, x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases} \quad (4)$$

$$f_{\text{sigmoid}}(x) = \frac{1}{1 + e^{-x}} \quad (5)$$

$$f_{\text{tanh}}(x) = \frac{2}{1 + e^{-2x}} - 1 \quad (6)$$

Swish is an activation function that includes a learnable parameter. Nearly all implementations do not use the learnable parameter, in which case the activation function is (“Swish-1”) in Eq. 7.

$$f_{\text{swish}}(x) = x * \text{sigmoid}(\beta x) \quad (7)$$

The SiLU (sigmoid linear unit), introduced before being popularized as “Swish,” was first mentioned alongside Gaussian error linear units (GELUs) and later explored in reinforcement learning research. It gained further attention through the study “Swish: A Self-Gated Activation Function,” highlighting its effectiveness in various neural network applications. This illustrates the SiLU/Swish function’s development and increasing acceptance in the field [45]. These three advanced activation functions are mathematically represented in Eqs. 8 and 9.

$$f_{\text{SiLU}}(x) = x * \text{sigmoid}(x) = \frac{x}{1 + e^{-x}} \quad (8)$$

$$f_{\text{GELU}}(x) = x\Phi(x) = x \left(1 + \text{erf}\left(\frac{x}{\sqrt{2}}\right) \right) / 2 \quad (9)$$

The Mish activation function outperforms ReLU in various tasks, enhancing accuracy in models from YOLOv1 to YOLOv5. Its integration into YoLov4 yielded a substantial qualitative improvement, exceeding the performance of YOLOv3 in mean average precision (mAP). Mish

distinguishes itself in the broader landscape of prevalent activation functions encompassing ReLU, Tanh, Sigmoid, Leaky ReLU, and Swish. In a specific instance involving Squeeze Excite Net-18 for CIFAR 100 classification, Mish demonstrated superior performance to Swish and ReLU, underscoring its distinctive advantages. The Mish activation function, expressed in Eq. 10, where Softplus(x) = $\ln(1 + e^x)$, offers a refined alternative to ReLU, consistently proving effective in enhancing model accuracy [46].

$$f_{\text{Mish}}(x) = x * \tanh[\text{Softplus}(x)] = x \cdot \tanh[\ln(1 + e^x)] \quad (10)$$

The dropout layer, a regularization layer, is used to eliminate specific node connections within the network to prevent it from excessively memorizing the trained data [47]. By randomly disabling network parameters, the process facilitates more effective learning. Control over the size of acquired features from the preceding layer is achieved by utilizing chosen filters that traverse the image, thus implementing a subsampling mechanism in the pooling layer. As a result, the model is safeguarded against overfitting, and the subsequent layer’s data is reduced. However, it is important to acknowledge that this reduction entails the loss of specific vital data. The pooling layer offers a significant advantage by reducing the number of parameters that necessitate computation in the network, thereby diminishing the network’s computational complexity and facilitating expedited training. Various techniques, including average, minimum, or maximum pooling, can be employed within the pooling process. Maximum pooling involves selecting the largest value among the pixel values within the filter size, while minimum pooling chooses the smallest value. Conversely, average pooling entails dividing the sum of pixel values within the filter size region by the filter window size. Figure 4 visually illustrates the processes of maximum, average, and minimum pooling.

Following the non-linear, pooling, and convolution layers, CNNs feature the fully connected layer (FC). This layer transforms output feature matrices into feature vectors by stacking individual vectors, resembling a grading process. The number of such layers may vary based on the design.

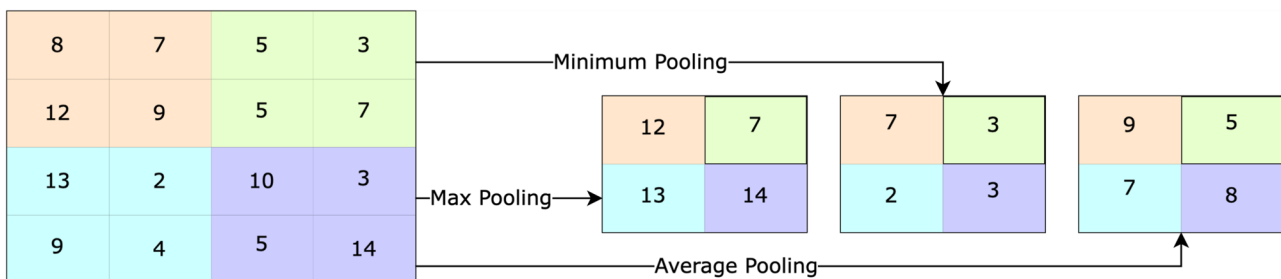


Fig. 4 Maximum, average, and minimum pooling

This resulting feature vector is subsequently labeled like a regular neural network. SoftMax has been one of the most preferred choices for classification tasks owing to its diverse accomplishments and classifications. The output for each object type falls within the [0,1] range. The object is assigned to a class for classification based on the output neuron with the highest value. Beyond the CNN layers, several crucial aspects depend on user experience throughout the training process. These aspects, known as hyperparameters, encompass maximum epoch, momentum, initial learning rate, minibatch size, regularization, and shuffle features.

Advanced CNN Architectures

CNNs have become the most used deep learning techniques, providing the highest throughput among other AI algorithms. Researchers have collaborated to propose new approaches for increasing the efficiency of CNNs. Different typical and recent Deep CNN architectures, which are now utilized as building blocks in various classification, segmentation, and detection designs, will be briefly discussed in this section. The earliest CNN model is the LeNet architecture, described in research by Yann LeCun [28]. This model,

fundamentally simple and basic, has a relatively small scale. This structure was used to classify handwritten pictures.

CNN launched in 2012 when the AlexNet design [29] won the ImageNet competition, comparable to the design of LeNet. It is a more robust structure with more feature filters. The AlexNet architecture success has altered the path of image processing research significantly. The ZFNet design [47] improves the AlexNet design’s architecture, leading the ImageNet competition in 2013 [48]. Figure 5 depicts the architecture of VGG16. Furthermore, the VGG architecture demonstrated a 7.3% error rate [49].

Later, several additional designs were introduced, including DenseNet [50], EfficientNet [32], and MobileNets [33]. VGG, Inception, and ResNet are the most often used architectures for disease analysis and image classification. In addition, while CNN structures were first employed for classification, they were widely adopted because of their effectiveness in object identification and segmentation applications. Many practical CNN-based algorithms were published in this sector since object identification is one of the most prominent bases of deep learning. Because of its success, faster R-CNN [51], one of these two-phase topologies in object detection, has been raised [30].

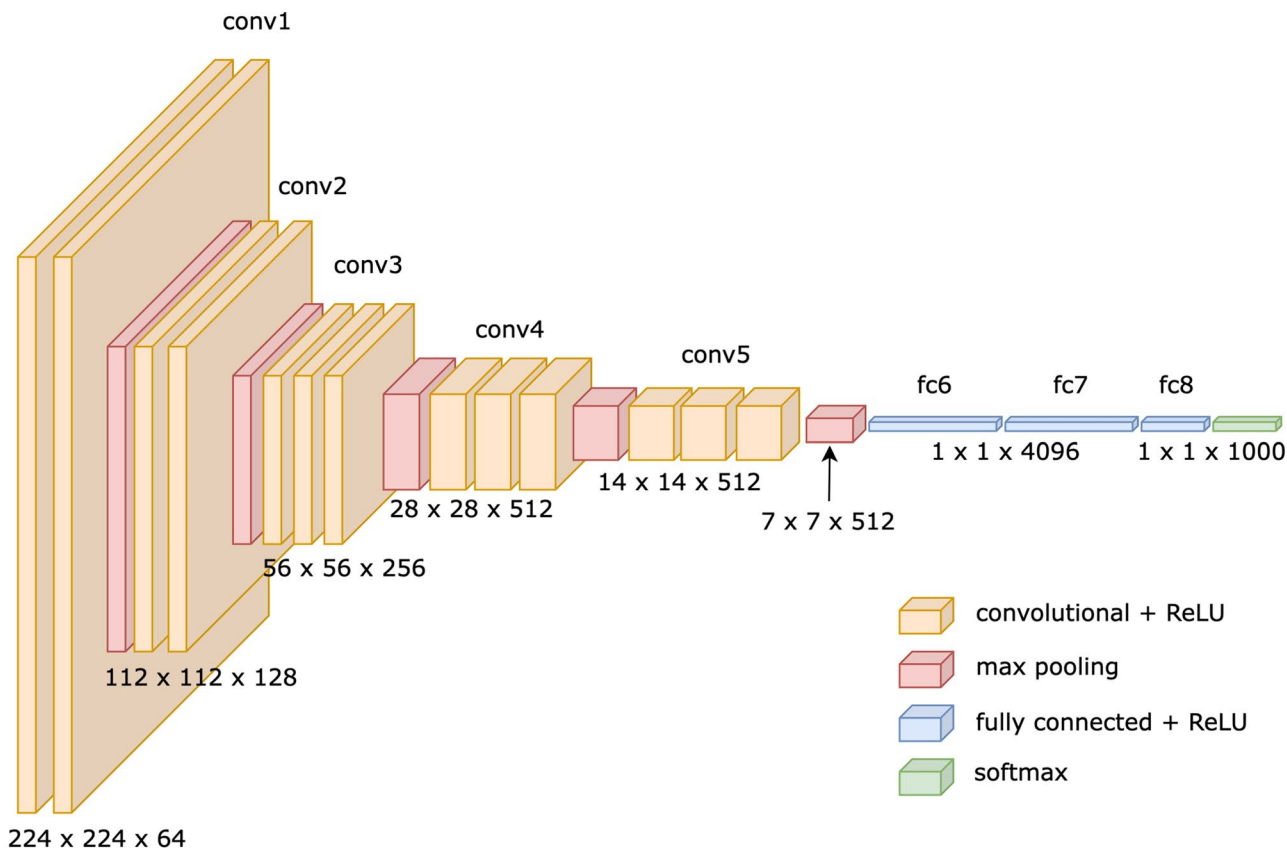


Fig. 5 Architecture of VGG16 network

VGG Architecture

The VGG network represents a major breakthrough in deep learning, especially in image recognition tasks. VGG architectures, as shown in Fig. 5, are deep CNN architecture that uses blocks consisting of only 3×3 dimensional filters and an increasing number of convolutional layers [52]. Also, to reduce the size of the resulting activation maps, the size is reduced by half by interspersing the maximum pooling blocks between the convolutional ones. Lastly, a classification block consists of two dense layers, each with 4096 neurons and the output layer with 1000 neurons. In VGG, the suffixes 16 and 19 indicate each network's respective number of weighted layers.

ResNet Architectures

Conversely, ResNet architectures gained prominence primarily due to the introduction of residual connections. Among these architectures, ResNet50 has garnered substantial popularity, boasting a significantly greater number of layers than VGG, yet requiring a mere one-fifth of the memory, thanks to including the global average pooling layer. In contrast, the initial version of Inception architectures manifests as a 22-layer deep CNN network comprising 5 million parameters. These architectures possess the ability to capture distinct features through the utilization of filters with dimensions of 1×1 , 3×3 , and 5×5 . Integrating 1×1 filters serves the purpose of size reduction, effectively eliminating computational bottlenecks [52]. Xception architectures, on the other hand, act as derivatives of Inception modules, having undergone notable modifications by incorporating deeply separable folds. Furthermore, several renowned

CNN architectures exist, including MobileNet, SqueezeNet, ResNet, and DenseNet. Each architecture harbors its own set of advantages and disadvantages, catering to different domains. For instance, architectures with fewer parameters and memory requirements are favored for real-time applications [15]. Conversely, deeper architectures are preferred for achieving heightened performance and facilitating diverse feature extractions, as depicted in Fig. 6 [53].

RegNet Architecture

ResNet and its variations have undoubtedly achieved remarkable outcomes across various computer vision tasks. Nonetheless, the architecture has a significant limitation: the absence of intermediate layer information communication within its building blocks. To overcome this drawback, a novel memory mechanism, known as a regulator module, is introduced to the ResNet [53].

This module extracts complementary features from the intermediate layers and facilitates feedback into the ResNet architecture. The regulator module effectively captures spatiotemporal information by incorporating convolutional recurrent neural networks, such as long short-term memories (LSTMs) or convolutional gated recurrent units (GRUs). This integration culminates in the birth of the regulated residual network (RegNet), which seamlessly integrates into any ResNet architecture. Experimental investigations conducted on three image classification datasets unequivocally demonstrate the superior performance of the RegNet architecture. Compared to the standard ResNet, the squeeze-and-excitation ResNet, and other state-of-the-art architectures [53], the RegNet exhibits remarkable capabilities, as depicted in Fig. 6.

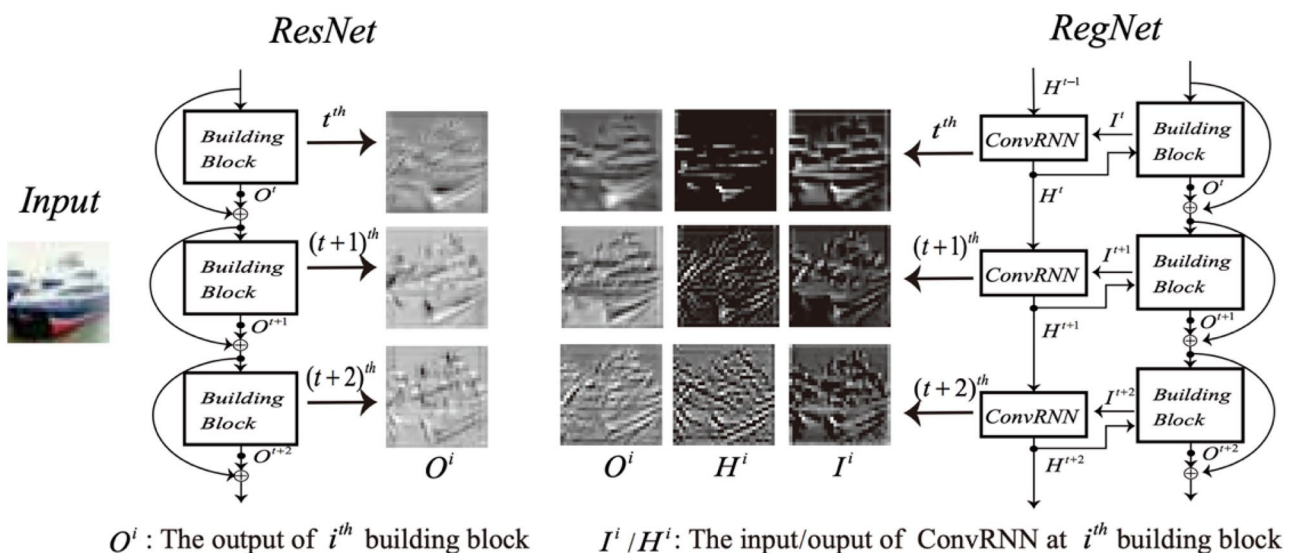


Fig. 6 ResNet and RegNet architecture and visualization of feature maps

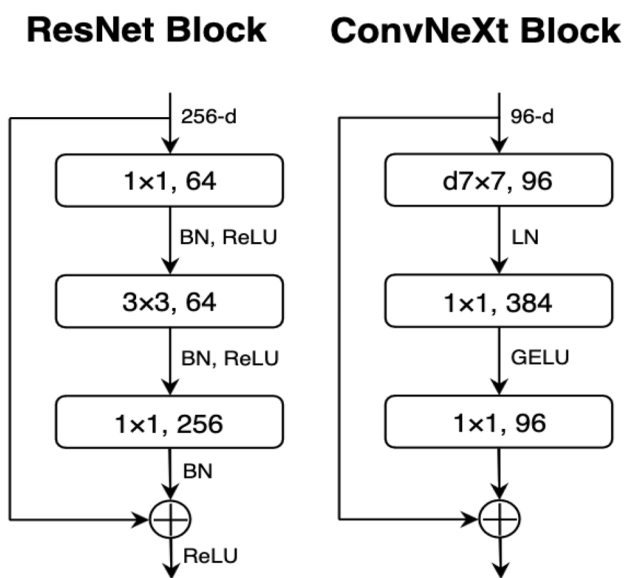


Fig. 7 Block designs for a ResNet and a ConvNeXt

ConvNeXt Architecture

The ConvNeXt architecture is a CNN architecture designed with visual recognition tasks in mind. Its primary focus is improving recognition accuracy, reducing computational costs, and increasing parallelism. The architecture employs a hybrid module that combines depthwise and group convolutions to achieve these goals. This hybrid module replaces the traditional stacked convolutional layers in other CNN architectures. ConvNeXt architecture can increase recognition accuracy while simultaneously reducing the computational cost of the model. The architecture also utilizes channel shuffle operations to increase the parallelism of the model. These operations allow the architecture to shuffle feature maps across channels, reducing the correlation between feature maps. This reduced correlation enables the architecture to parallelize more efficiently, resulting in faster and more accurate predictions. ConvNeXt has achieved state-of-the-art performance on several recognition benchmarks, including ImageNet classification, COCO detection, and ADE20K segmentation block designs comparison for a ResNet and a ConvNeXt presented in Fig. 7 [54]. This success is partly attributed to the architecture’s innovative design paradigm

and hybrid module, enhancing recognition accuracy, reducing computational costs, and increasing parallelism [54].

DenseNet Architecture

DenseNet, an acronym for dense convolutional network, marks a groundbreaking development in the deep learning landscape, particularly in image classification. Its unique approach involves directly connecting every layer to all subsequent layers in a feed-forward fashion, setting it apart from conventional architectures. Such dense connections enhance feature recycling, reduce the overall parameter count, and bolster the information transmission across the network. These attributes enable DenseNet to achieve outstanding efficiency and precision across diverse image recognition tasks, cementing its role as a crucial technology in the field of computer vision. Additionally, DenseNets significantly reduces the number of parameters, thus enhancing computational efficiency. Extensive evaluations are conducted on four demanding object recognition benchmarks, namely CIFAR-10, CIFAR-100, SVHN, and ImageNet, to assess the efficacy of the DenseNet architecture. The results demonstrate substantial improvements over the state-of-the-art methods across most of these benchmarks while demanding fewer computational resources to achieve exceptional performance shown in Fig. 8 [50].

Transformers Architectures

Transformers are a revolutionary deep learning method that has significantly impacted various fields, including natural language processing (NLP) and computer vision. They are neural network models designed to process sequential data by leveraging attention mechanisms. In healthcare, transformers have been applied to medical image analysis, electronic health records, and disease outcome prediction, showcasing their potential to extract meaningful insights from complex healthcare data. Vision transformers (ViTs) adapt the classical transformer architecture, making them powerful models for computer vision tasks. By emulating the structure and functionality of the human brain, transformers have revolutionized artificial intelligence by enhancing the accuracy and performance of various applications, albeit demanding substantial computational resources [55].

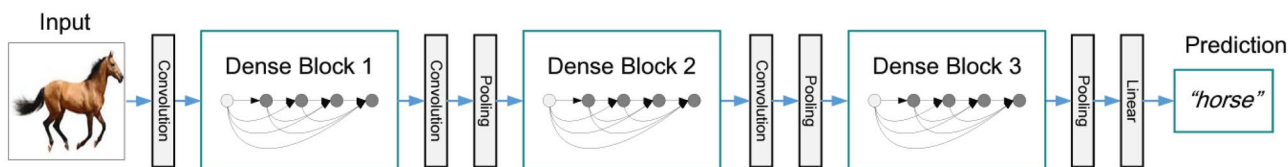


Fig. 8 A deep DenseNet architecture that is composed of three dense blocks

The vision transformer (ViT) has been applied to various healthcare contexts [56]. The latest developments in machine learning, including deep learning, provide a great opportunity to improve patient outcomes, reduce socioeconomic disparities, and enable early indication and detection of diseases in healthcare [57]. As a key component of deep learning, transformers have the potential to contribute to these advancements due to their ability to capture long-range dependencies and contextual information, making them suitable for analyzing complex healthcare data [11].

Dataset Preparation

Data Collection

This research was carried out jointly in the Department of Oral and Maxillofacial Radiology, Faculty of Dentistry, Erciyes University, and the Department of Computer Engineering, Engineering Faculty, Erciyes University, with the approval of the Erciyes University Ethics Committee. A total of 14,811 panoramic images of six implant brands were obtained from the Department of Oral and Maxillofacial Radiology, Faculty of Dentistry, Erciyes University. Ethical approval of the study was obtained from Erciyes University Ethics Committee number 2021/234 and followed by institutional guidelines. The archives of the Department of Oral and Maxillofacial Surgery, Faculty of Dentistry, Erciyes University were scanned to create a data pool from panoramic images. The file numbers of the patients who underwent implant surgery and the implant information were obtained. The hospital's file records and patient information system (MEDDATA) contained information about the

implant's brand, diameter, and bone or tissue level placed on the patient. Then, the panoramic radiographs taken routinely after the implant surgery of the patients in the faculty were accessed using the patient information system.

These radiographs were archived in JPEG format according to file numbers, creating a data pool. By creating separate sub-folders for each brand in the data pool, the status of the implant's bone or tissue level and its diameters were filed separately, and the data pool was created. The information obtained in the data pool was evaluated in three main parameters: the patient's implant brand, the tissue or bone level information, and the implant's diameter. The collected data were classified, and radiographs of implants that did not match the archive information or were inaccurate were removed from the data pool. Similarly, radiographs unsuitable for radio diagnostics were excluded from the collection. Information on six different implant brands is presented in Fig. 9 in detail. Images with serious noise, blur, distortion, and other conditions preventing the expert from marking will be removed from the dataset.

Radiographs with unidentifiable implants were removed, leaving 1258 implant images, rich in detail regarding brand, system, and diameter, for the project. These images underwent necessary preprocessing to prepare them for deep learning models. To achieve an even distribution across brands and assess the models' ability to generalize, the dataset was segmented into training, validation, and testing groups, as illustrated in Fig. 9. Some sample images from the dataset containing images of six implant brands are depicted in Fig. 10 [35]. Subsequently, these images were sorted into folders and uploaded to a labeling interface for thorough examination.

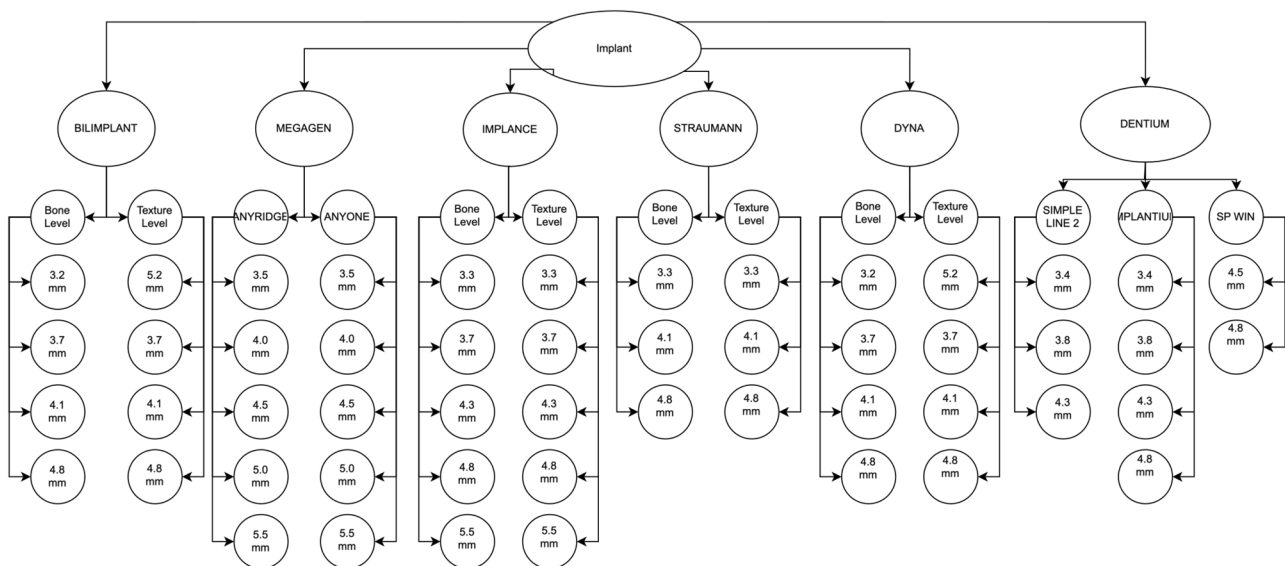
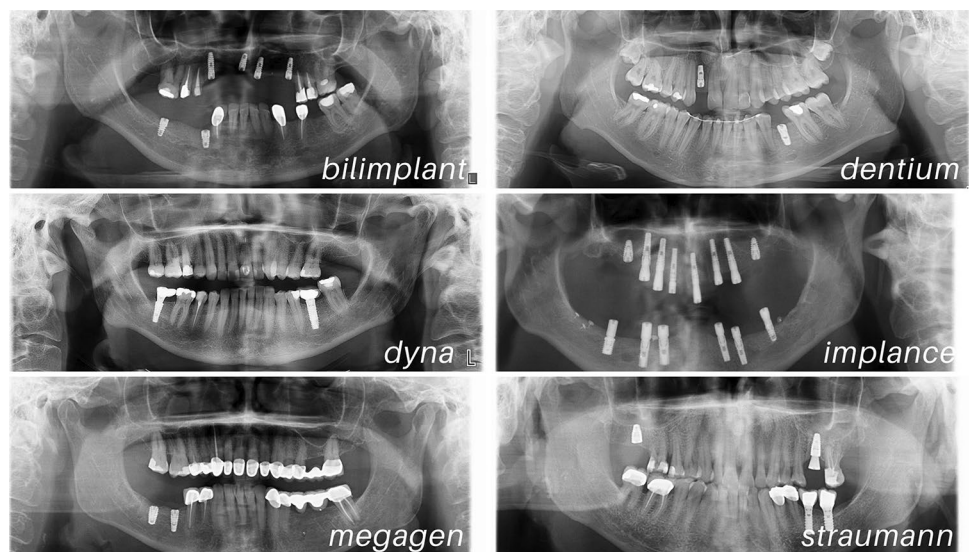


Fig. 9 Implant brands, systems, and diameters defined in the study

Fig. 10 The study used six different samples of implant brands in the dataset [35]



Data Annotations

Implants were labeled according to archival data directly on the actual image, without taking cross-sections from the panoramic radiographs in a single patient's mouth, where images also included different implant brands. The tagging interface developed by the researchers was used, and all data was stored in the cloud system of the same interface to prevent data loss. While labeling, the labeling rules in similar studies in the literature were followed, and the label was chosen as a rectangle [58]. With the help of this method, the screw part of the implant was entirely inside this rectangle, and the rectangle was kept as narrow as possible. In the labeling interface, the label was created from 2 points as a starting and ending rectangle, and each brand's name, the tissue/bone level options, and its diameter were determined. All this information was registered to the system interface, and the labels were labeled per the archive data from these records.

Data Preprocessing

In the preparation of the dataset for deep learning model training, specialist physicians played a critical role by meticulously reviewing and approving the labels for the implant systems, ensuring the annotations' accuracy and reliability. Following this expert verification, the dataset underwent a comprehensive preprocessing stage, where image processing techniques were employed to enhance image quality and remove noise. This step was vital for refining the dataset, utilizing methods such as noise filtering and contrast enhancement to eliminate irrelevant details and improve the visibility of crucial features.

Table 1 illustrates the distribution of 1258 implant images across six brands for deep learning model training, including Straumann (106), Bilimplant (121), Impliance (236), Dyna (248), Megagen (253), and Dentium (294). As part of data preprocessing, images are resized to 224×224 pixels to standardize input size. Further preprocessing includes normalization to adjust pixel values for computational efficiency and augmentation to introduce variabilities, such as rotations and flips. These steps are crucial for preparing the dataset, ensuring models are trained on consistent, diverse data for accurate implant brand recognition.

Data Augmentation

In this study, we implemented various data augmentation techniques to improve the performance and generalization ability of deep learning models that require a large amount of data. We utilized the most recent data augmentation methods available in the PyTorch library. The data augmentation techniques used in this study are listed in Table 2 and were applied during the training process rather than before training.

Table 1 Implant brands and numbers used in this study

Implant brand	Total
Straumann	106
Bilimplant	121
Implance	236
Dyna	248
Megagen	253
Dentium	294
Total images	1258

Table 2 Some hyper-parameters and their values employed for data augmentation in this research

Hyper-parameter	Value	Hyper-parameter	Value
Hue (fraction)	0.015	Scale (+/- gain)	0.5
Rotation (+/- degrees)	0.1	Flip (up-down, right-left, probability)	0.5

One of the main benefits of using data augmentation during training is that it allows each image to be used only once with a specific probability, which reduces the training time for the model. To ensure a fair comparison, we used default values rather than optimizing hyperparameters, and examples of augmented images can be seen in Fig. 11 [35].

Proposed Decision Support System

In this study, we proposed a deep learning-based diagnostic system for the autonomous identification of implant brands, incorporating a combination of 18 current CNN models and 10 vision transformer models that have gained significant popularity in recent years. As illustrated in Fig. 12 [35], the methodology encompasses the creation of a novel dataset meticulously labeled by experts (at least four dental specialists in the field), training deep learning models, applying transfer learning (utilizing pre-trained ImageNet weights), and employing data augmentation techniques. This comprehensive approach aims to test and identify the most optimal model for effective and accurate implant brand recognition.

The dataset at our disposal was partitioned into training, validation, and test subsets, with the former dedicated to training CNN architectures. Additionally, we delved into transfer learning, leveraging pre-trained models from various

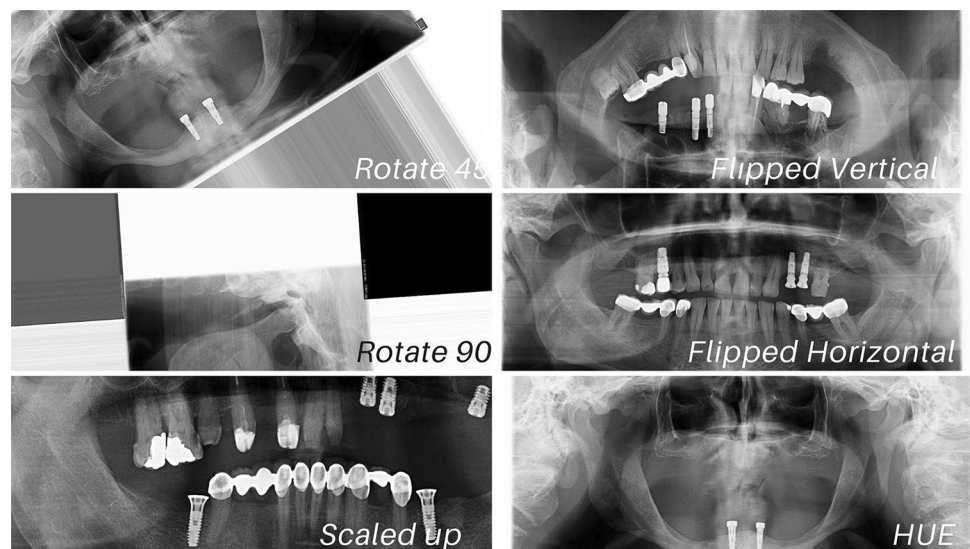
datasets to enhance our approach. Upon completing the training phase, we pinpointed the most effective network through its test set performance, incorporating insights from domain-specific medical experts. Classification accuracy, the ratio of correctly identified cases, was our principal metric for gauging performance. We also utilized secondary metrics, including examining features detected by the CNN within pertinent image areas. We calculated several key metrics to assess the classifier's efficacy on the test dataset: accuracy, precision, recall, F1-score, and the area under the ROC curve, using a confusion matrix. These evaluations considered the congruence between the actual labels of positive dental implants and the classifier's predictions, providing a comprehensive overview of the system's diagnostic capabilities.

Experiments and Discussion

Implementation Details

The study included specific system specifications, including Ubuntu 20.04 as the operating system. The processing power relied on an Intel Core i9 9900X processor, featuring ten cores and a clock speed of 3.50 GHz. The plan had 32 GB of DDR4 RAM to ensure optimal performance. Additionally, an RTX 2080TI graphics card, with 11 GB of GDDR6 memory and 4352 CUDA cores, was employed. The study used various software components, including Python 3.8, PyTorch, NVIDIA CUDA Toolkit 11.4, and NVIDIA GPU-Accelerated Library (cuDNN) version 8.2. Several performance metrics, such as classification accuracy, a confusion matrix, and a classification report, were employed to assess the model's efficacy. Classification accuracy was calculated

Fig. 11 Samples of techniques used on implant system dataset augmentation [35]



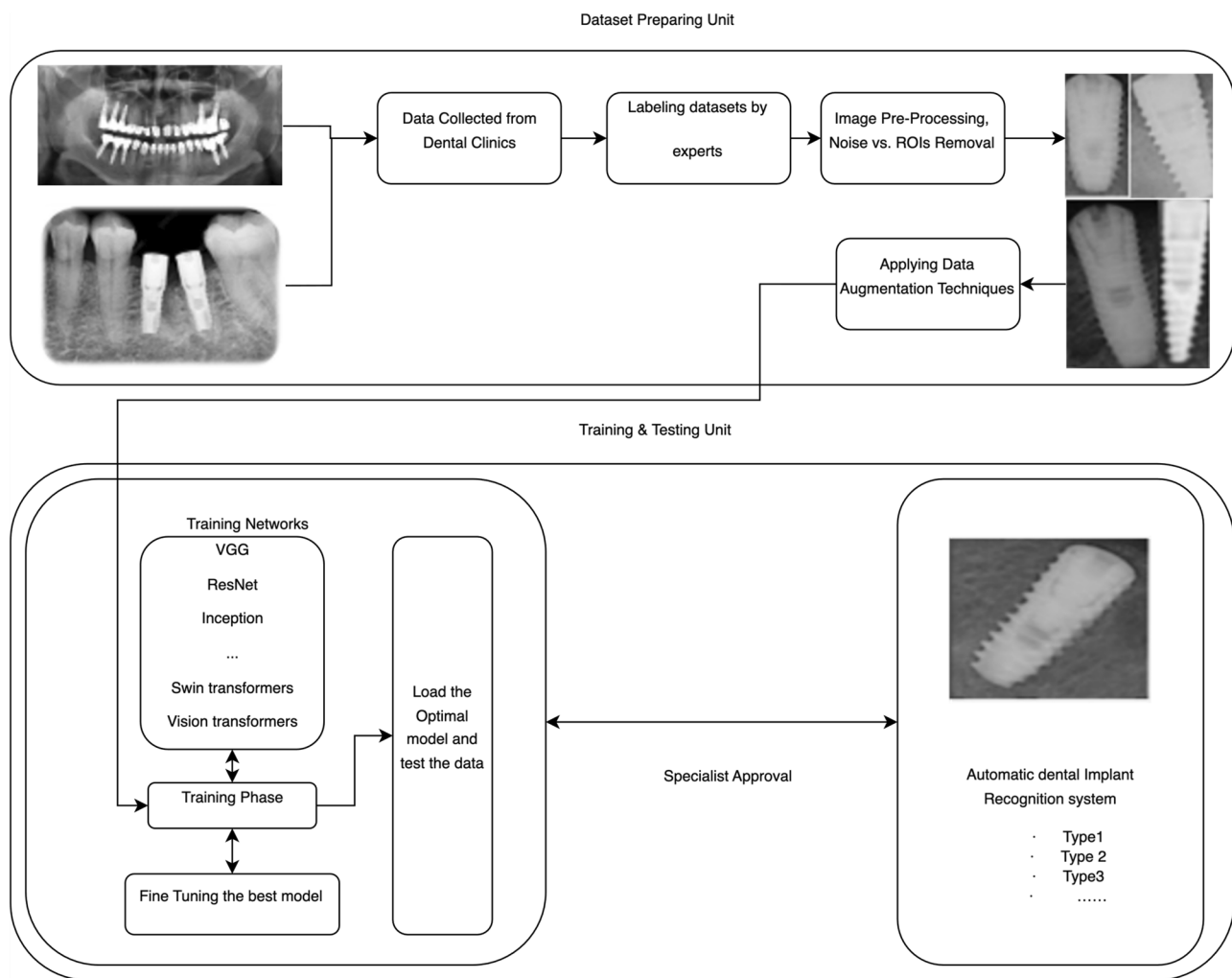


Fig. 12 Proposed decision support system architecture [35]

by dividing the number of correct predictions by the total number of input samples. However, it is crucial to note that relying solely on overall classification accuracy may create a misleading impression of high accuracy, particularly when misclassifying samples from the minor class is more probable. Consequently, this study incorporated additional evaluation metrics to scrutinize the accuracy of the implemented classifier thoroughly.

Transfer Learning (TL)

When tackling the task of training a CNN model for medical image analysis, a significant hurdle arises in acquiring large-scale labeled datasets. However, this limitation can be overcome by employing transfer learning (TL), which capitalizes on the learned parameters (i.e., weights) of a well-established CNN model trained on a substantial dataset like ImageNet. This can be accomplished by fine-tuning

or freezing the convolutional layers of the pre-trained CNN model and initiating the training of the fully connected layers from scratch, utilizing the medical dataset. The essence of transfer learning lies in understanding that fundamental features, such as straight lines and curves that form the basis of images, possess universal applicability across diverse image analysis tasks. Consequently, the transferred weights serve as a dependable set of features, diminishing the reliance on extensive datasets while minimizing training time and memory requirements. Two distinct approaches exist within transfer learning: feature extraction and fine-tuning [59].

Evaluation Metrics

To ensure accurate and reliable results, the classification accuracy report must encompass several essential metrics, including precision, recall, F1, and support scores for each class within the model. These metrics play a crucial role in

assessing the accuracy and effectiveness of the classifier. Precision, in particular, is a key indicator of a classifier's accuracy. It calculates the ratio of true positives to the total number of true positives and false positives. By referring to Eq. 11, one can determine the percentage of correctly classified positive cases out of all the positive cases present.

$$\text{Precision} = TP / (TP + FP) \quad (11)$$

The measure of recall serves as an indicator of the classifier's completeness and capacity to identify all positive instances correctly. It quantifies the ratio of true positives to the total number of true positives and false negatives for each class, as demonstrated in Eq. 12. This metric provides insights into the classifier's effectiveness in capturing and retrieving all relevant positive instances within the dataset.

$$\text{Recall} = TP / (TP + FN) \quad (12)$$

The F1-score, ranging from 0.0 to 1.0, represents a weighted harmonic average of precision and recall. It comprehensively assesses a classifier's performance, considering its ability to correctly identify positive instances (precision) and its completeness in capturing all positive instances (recall). The F1-scores are typically lower than other accuracy measures as they incorporate precision and recall. Instead of relying solely on overall accuracy, the weighted average of F1-scores is often employed to evaluate the effectiveness of classifier models, as depicted in Eq. 13. This approach offers a more robust and nuanced evaluation of the classifier's performance, considering the trade-off between precision and recall.

$$\text{F1Score} = 2 * (\text{Recall} * \text{Precision}) / (\text{Recall} + \text{Precision}) \quad (13)$$

The following four values represent this:

- True positive (TP): When an element is predicted to be positive, and it is.
- False positive (FP): An element is expected to be positive but not.
- True negative (TN): An element is expected to be negative, and it is.
- False negative (FN): An element is expected to be negative, but it is positive.

Evaluating Training Strategy

In this study, the effectiveness of our training approach was assessed through the use of the stochastic gradient descent (SGD) optimizer, with the inclusion of defined parameters. These parameters included a learning rate (lr) of 0.001, a momentum of 0.9, and a learning rate scheduler comprising a step size of 7 and a gamma value of 0.1. The SGD optimizer is widely employed in deep learning models, particularly for image classification tasks. The learning rate governs the extent of parameter updates during training, while momentum aids in avoiding local minima and guiding the model toward the global minimum. A learning rate scheduler was implemented to enhance model performance, adjusting the learning rate throughout training. A step size of 7 was selected, indicating that the learning rate would decrease by 0.1 every seven epochs. This approach facilitated quicker convergence and mitigated overfitting. Table 2 highlights the various data augmentation techniques employed to augment the training data and alleviate overfitting. The model was trained for 300 epochs, utilizing cross-entropy loss as the objective function. The performance of our training strategy was assessed using the same metrics as outlined in the "Evaluation Metrics" section, including accuracy, precision, recall, and F1-score.

RegNet Architectures

The performance metrics of fine-tuned RegNet architectures, both before and after applying data augmentation, are presented in Table 3. The two models under evaluation are regnet_y_32gf and regnet_y_16gf. These models were assessed using various performance metrics, including accuracy, precision, recall, and F1-score, which provide insights into their ability to classify images accurately. Before data augmentation, regnet_y_32gf achieved an accuracy of 0.904, precision of 0.896, recall of 0.89, and an F1-score of 0.888. After incorporating data augmentation, the accuracy improved to 0.936, precision to 0.928, recall to 0.928, and F1-score to 0.928. Similarly, regnet_y_16gf achieved an accuracy of 0.872, precision of 0.872, recall of 0.844, and an F1-score of 0.840 before data augmentation. Following data augmentation, the accuracy increased to 0.924, precision to 0.93, recall to 0.927, and F1-score to 0.934. These results highlight the

Table 3 Performance metrics of fine-tuning of RegNet architectures

Model\average	Without applying data augmentation				With applying data augmentation			
	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
regnet_y_32gf	0.904	0.896	0.89	0.888	0.936	0.928	0.928	0.928
regnet_y_16gf	0.872	0.872	0.844	0.840	0.934	0.924	0.93	0.927

Table 4 Performance metrics of fine-tuning of DenseNet architectures

Model\average	Without applying data augmentation				With applying data augmentation			
	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
Densenet-121	0.854	0.832	0.832	0.828	0.932	0.934	0.916	0.922

benefits of data augmentation for both models, as evidenced by the accuracy, precision, recall, and F1-score improvements. Furthermore, the extent of enhancement in F1-score accuracy can be quantified. For RegNet_y_32gf, the average improvement is 4.00%, while for RegNet_y_16gf, it reaches 8.70%. This notable increase underscores the substantial performance boost achieved through data augmentation. These findings affirm the value of data augmentation as a technique to enhance the performance of deep learning models in image classification tasks. It is important to note that the impact of data augmentation on performance was more pronounced for regnet_y_32gf compared to regnet_y_16gf.

DenseNet Architectures

The performance of the DenseNet-121 architecture, as depicted in Table 4, reveals a remarkable enhancement in the model's effectiveness through data augmentation. This is evident from the substantial increase in accuracy, amounting to 7.8%. Moreover, there is a notable improvement in precision, which rose from 0.832 to 0.934, representing an increase of 10.2%. Similarly, the recall increased from 0.832 to 0.916, signifying a noteworthy improvement of 8.4%. Additionally, the F1-score exhibited a significant increase, rising from 0.828 to 0.922, reflecting an improvement of 9.4%. These results underline the considerable positive impact of data augmentation on the performance of the DenseNet-121 architecture.

VGG Architectures

The performance metrics of VGG models are presented in Table 5, showcasing the impact of applying data augmentation. Notably, VGG16 exhibited notable enhancements across multiple metrics, with accuracy increasing to 0.926, precision to 0.932, recall to 0.905, and F1-score to 0.918. Similarly, VGG19 demonstrated improvements in accuracy (0.924), precision (0.92), recall (0.91), and F1-score (0.915).

Table 5 Performance metrics of fine-tuning of VGG architectures

Model\average	Without applying data augmentation				With applying data augmentation			
	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
VGG16	0.899	0.897	0.880	0.888	0.926	0.932	0.905	0.918
VGG19	0.894	0.888	0.868	0.878	0.924	0.92	0.91	0.915

These findings substantiate the significant role of data augmentation in enhancing the performance of deep learning models when tackling image classification tasks.

Efficientnet Architectures

The performance metrics of fine-tuning Efficientnet architectures, both before and after incorporating data augmentation, are presented in Table 6. Before data augmentation, Efficientnet_b4 exhibited an accuracy of 0.882 and an F1-score of 0.860. However, after integrating data augmentation techniques, these metrics experienced significant improvement, with the accuracy rising to 0.924 and the F1-score increasing to 0.916. Similarly, Efficientnet_b0 showcased an accuracy of 0.880 and an F1-score of 0.858 before data augmentation. Nevertheless, after the implementation of data augmentation, these metrics were enhanced, resulting in an accuracy of 0.886 and an F1-score of 0.874. Consequently, the performance of both Efficientnet models was substantially improved through data augmentation.

ResNet Architectures

Based on the performance metrics of fine-tuning of ResNet Architectures shown in Table 7, we can observe that ResNet50, wide_resnet101_2, and wide_resnet50_2 have good accuracy scores of 0.894, 0.874, and 0.888, respectively, after applying data augmentation. These architectures also show improvements in other metrics such as precision, recall, and F1-score. On the other hand, ResNet18 and ResNet34 have lower accuracy scores but still show improvements in the precision, recall, and F1-scores after applying data augmentation. ResNet101 has a higher accuracy score than ResNet18 and ResNet34 but a lower score than ResNet50 and the wide ResNet models. Therefore, based on these results, ResNet50, wide_resnet101_2, and wide_resnet50_2 could be good choices for this dental implant brand classification task.

Table 6 Performance metrics of fine-tuning of Efficientnet architectures

Model\average	Without applying data augmentation				With applying data augmentation			
	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
Efficientnet_b4	0.882	0.88	0.86	0.860	0.924	0.924	0.908	0.916
Efficientnet_b0	0.88	0.872	0.856	0.858	0.88	0.886	0.7096	0.874

ConvNext Architectures

The impact of data augmentation on the performance of both ConvNeXt architectures is evident in Table 8. Notably, data augmentation played a pivotal role in improving the accuracy of the large model from 0.788 to 0.878, while the base model's accuracy increased from 0.745 to 0.856. Moreover, both models observed significant enhancements across various metrics, including precision, recall, and F1-score. Specifically, the large model's precision, recall, and F1-score experienced substantial improvements, rising from 0.776, 0.732, and 0.724 to 0.884, 0.842, and 0.85, respectively. Similarly, the base model showcased noteworthy enhancements in precision, recall, and F1-score, with values progressing from 0.743, 0.712, and 0.719 to 0.854, 0.818, and 0.826, respectively. These findings indicate that data augmentation has significantly bolstered the performance of both ConvNeXt architectures across all evaluated metrics, highlighting its effectiveness in improving the models' classification capabilities.

MobileNet Architectures

Table 9 presents the performance metrics of two MobileNet architectures before and after applying data augmentation. Both models have demonstrated improvement in their performance after applying data augmentation. The MobileNet_v2 architecture achieved an accuracy score of 0.8 before data augmentation, which significantly increased to 0.844 after data augmentation. The MobileNet_v2 architecture increased precision, recall, and F1-score from 0.76, 0.752, and 0.748 to 0.834, 0.816, and 0.824, respectively. Overall, the findings suggest that applying data

augmentation has positively impacted the performance of both MobileNet architectures.

Transformers Architectures

The performance metrics of several transformer architectures before and after applying data augmentation are presented in Table 10, including accuracy, precision, recall, and F1-score. The results indicate that all models experienced performance improvements after data augmentation. The maxvit_large and maxvit_small models demonstrated a notable increase in accuracy score, rising from 0.839 and 0.833 before data augmentation to 0.930 and 0.926 after. Conversely, the accuracy of the swin_large and swin_small models improved from 0.823 to 0.915 and from 0.785 to 0.872, respectively, after data augmentation. For the vit_b_16 model, the accuracy score improved from 0.819 to 0.910 after data augmentation, while the vit_b_32 model improved from 0.785 to 0.872. The vit_l_32 model improved from 0.771 to 0.856 before and after data augmentation. In conclusion, data augmentation enhanced performance for the swin_large, swin_small, vit_b_16, and vit_b_32 models, while the swin_s and vit_l_32 models did not experience significant improvements.

Discussion

Performance metrics from Table 3 to 9 reveal that data augmentation significantly enhances the accuracy, precision, recall, and F1-score across many models. This technique, which enlarges and diversifies the training dataset, helps mitigate overfitting and boosts overall model efficacy.

Table 7 Performance metrics of fine-tuning of ResNet architectures

Model\average	Without applying data augmentation				With applying data augmentation			
	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
ResNet50	0.83	0.82	0.802	0.802	0.894	0.886	0.886	0.884
wide_resnet101_2	0.822	0.814	0.796	0.794	0.874	0.89	0.874	0.88
wide_resnet50_2	0.836	0.82	0.818	0.814	0.888	0.888	0.868	0.876
ResNet101	0.808	0.798	0.782	0.786	0.868	0.866	0.86	0.864
ResNet34	0.782	0.78	0.75	0.75	0.844	0.836	0.824	0.824
ResNet18	0.748	0.724	0.708	0.698	0.82	0.808	0.786	0.79

Table 8 Performance metrics of fine-tuning of ConvNeXt architectures

Model\average	Without applying data augmentation				With applying data augmentation			
	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
ConvNeXt (tiny)	0.847	0.853	0.835	0.842	0.942	0.948	0.929	0.936
ConvNeXt (small)	0.847	0.860	0.839	0.847	0.942	0.956	0.933	0.942
ConvNeXt (large)	0.788	0.776	0.732	0.724	0.878	0.884	0.842	0.850
ConvNeXt (base)	0.745	0.743	0.712	0.719	0.856	0.854	0.818	0.826

Table 9 Performance metrics of fine-tuning of MobileNet architectures

Model\average	Without applying data augmentation				With applying data augmentation			
	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
MobileNet_v2	0.8	0.76	0.752	0.748	0.844	0.834	0.816	0.824

Table 10 Performance metrics of fine-tuning of transformers architectures

Model\average	Without applying data augmentation				With applying data augmentation			
	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
swin_large	0.823	0.830	0.816	0.822	0.915	0.922	0.906	0.913
swin_small	0.785	0.777	0.767	0.770	0.872	0.863	0.853	0.856
vit_b_16	0.819	0.839	0.809	0.818	0.910	0.9327	0.8994	0.909
vit_b_32	0.785	0.782	0.777	0.778	0.872	0.8693	0.8638	0.864
vit_l_16	0.829	0.844	0.811	0.822	0.920	0.938	0.901	0.913
vit_l_32	0.771	0.763	0.766	0.764	0.856	0.848	0.851	0.848
maxvit_tiny	0.814	0.824	0.799	0.807	0.904	0.916	0.888	0.897
maxvit_small	0.833	0.829	0.821	0.823	0.926	0.921	0.913	0.915
maxvit_base	0.814	0.815	0.813	0.812	0.904	0.906	0.903	0.902
maxvit_large	0.839	0.842	0.837	0.839	0.930	0.935	0.93	0.932

ConvNeXt architectures stood out in classifying dental implant systems, particularly benefiting from data augmentation. The ConvNeXt_small model led the pack with

a remarkable accuracy of 94.2%, precision at 95.6%, recall of 93.3%, and an F1-score of 94.2%, closely followed by the ConvNeXt_tiny variant. RegNet models, especially

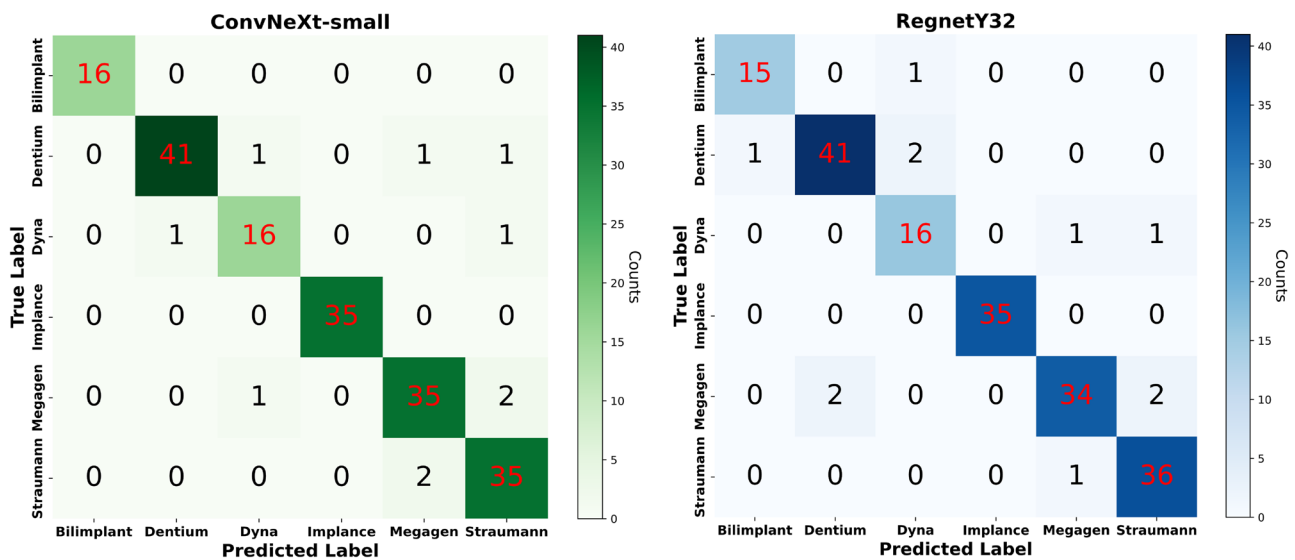


Fig. 13 Confusion matrices of ConvNeXt-small and RegNetY32 models, achieving optimal results for dental implant identification


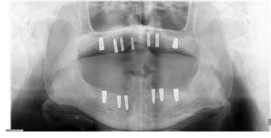

















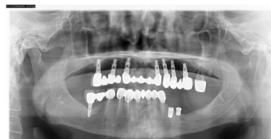
	Correct Classification		Incorrect Classification	
Bilimplant	 Target: Bilimplant	 Prediction: Bilimplant	For Bilimplant all test images were correctly classified.	
Dentium	 Target: Dentium	 Prediction: Dentium	 Target: Dentium	 Prediction: Dyna
Dyna	 Target: Dyna	 Prediction: Dyna	 Target: Dyna	 Prediction: Dentium
Implance	 Target: Implance	 Prediction: Implance	For Implance all test images were correctly classified.	
Megagen	 Target: Megagen	 Prediction: Megagen	 Target: Megagen	 Prediction: Dyna
Straumann	 Target: Straumann	 Prediction: Straumann	 Target: Straumann	 Prediction: Megagen

Fig. 14 Examples of false negatives and false positives. Bilimplant, Dentium, Dyna, Implance, Megagen; Straumann

RegNet_y_32gf alongside ConvNeXt_small, showed significant improvement across all metrics with data augmentation. Similarly, DenseNet-121 and VGG models experienced notable performance boosts, with DenseNet-121 showing a significant jump in accuracy from 85.4 to 93.2%. The EfficientNet_b4 model also enhanced performance with data augmentation over the EfficientNet_b0 model.

In contrast, ResNet18 lagged, indicating better options exist for classifying dental implant systems. Summarizing, ConvNeXt and RegNet architectures emerge as the top choices for this classification task, offering the highest accuracy, precision, recall, and F1-score. Figure 13 presents the confusion matrix for the model with the highest accuracy from each architecture.

These models demonstrated significant performance improvements with data augmentation, highlighting the effectiveness of this technique in enhancing the performance of deep learning models. We demonstrated the effectiveness of fine-tuning and data augmentation techniques on various architectures. ConvNeXt models perform the highest in classifying six dental implants based on panoramic X-ray images. The ConvNeXt-small algorithm had the best classification accuracy and F1-score performance. The model resulted in the impeccable classification of Bilimplant and Implants classes, achieving a classification accuracy of 100%. Moreover, Strauman exhibited an approximate accuracy rate of 95%, Dentium manifested a precision of 93%, Megagen demonstrated a proficiency of 92%, and Dyna registered an accuracy rate of 88%. The graphical representation of these outcomes employs the x-axis for predicted labels and the y-axis for true labels within the confusion matrices. Notably, the matrix rows delineate instances of false negatives, while the columns delineate occurrences of false positives, providing a comprehensive visualization of the model's classification performance, as shown in Fig. 14.

Examining the confusion matrix depicted in Fig. 13 and evaluating the detections of the ConvNeXt-small model on the test set in Fig. 14, it is apparent that the model accurately identifies the Bilimplant and Implants classes. However, there are more incorrect detections observed in other classes. This phenomenon can be attributed to the fact that deep learning algorithms typically exhibit greater efficacy with larger datasets. Despite the dataset containing a limited number of training images for each implant brand class, totaling 1258 images, the model may encounter difficulty in distinguishing between similar implant brands. As illustrated in Fig. 14, it is evident that classes which are often closely related, such as Dyna and Dentium, are misidentified. Working with larger datasets and utilizing finely-tuned models, along with employing different data augmentation techniques, can lead to more favorable outcomes in the effective detection of implant classes. While these models achieve high accuracy with a small-scale dataset, it is evident that employing more effective techniques and larger datasets can result in higher levels of success. Therefore, we hope to achieve more accurate results in identifying implant brands in the future by utilizing better-optimized, more advanced, and effective models.

Limitations and Future Work

This study encountered several limitations that warrant acknowledgment and consideration. Firstly, the retrospective nature of the investigation introduced a potential spectrum bias, although all dental radiographic images were sourced from the same hospital. This inherent limitation implies that the study may not fully represent the broader population or diverse clinical scenarios. Furthermore, a notable area for

enhancement pertains to the automated deep-learning algorithm employed in this study. It did not undertake an analysis or evaluation of periapical radiographic images; consequently, these crucial images were excluded from the scope of the investigation. This omission represents a significant limitation as periapical images are recognized in previous research as being instrumental for accuracy in assessing apical and periodontal conditions. The exclusion of periapical radiographs might have implications on the comprehensiveness of the findings, and future studies should consider incorporating such images to ensure a more comprehensive and representative evaluation of dental conditions. In summary, while this study provides valuable insights, the mentioned limitations underscore the need for cautious interpretation of the results. Future research should address these constraints and adopt a more inclusive approach to cover a broader range of dental imaging data, including periapical radiographs, to enhance the overall robustness and generalizability of the findings.

Conclusion

In this study, we highlighted the impressive capability of various deep learning models to accurately classify six different dental implant systems using panoramic X-ray images. These models showcased their potential for deep learning, even in the face of diverse conditions encountered during the implant treatment phase. By employing transfer learning and fine-tuning techniques on pre-trained deep CNN architectures, we achieved high accuracy in image classification, overcoming the challenges posed by a relatively modest and imbalanced dataset. The outcomes of this study provide a valuable reference point for future research endeavors in dental implant identification. Notably, the ConvNeXt_small model exhibited exceptional proficiency in classifying dental implant brands, effectively addressing issues associated with dataset imbalances. The deep learning architectures demonstrated classification accuracy on panoramic radiographs comparable to board-certified periodontists. These advancements hold significant potential for dental professionals in their clinical practice, offering an efficient and precise method for classifying diverse dental implant systems. The study's findings underscore the applicability and promise of deep learning techniques in the field, paving the way for enhanced diagnostic capabilities in dental implantology.

Acknowledgements The computational experiments were conducted utilizing the resources available at the Artificial Intelligence and Big Data Application and Research Center at Erciyes University in Turkey. Ethical clearance for this study was acquired from Erciyes University Dental Hospital under the authorization of permission decision 2021/234, dated 24.03.2021. We express our gratitude to TÜBİTAK for their support of this project, which bears project number 121E068.

Author Contribution Mohammed A. H. Lubbad: conceptualization, methodology, software, reviewing, investigation, validation, data curation, writing—review and editing; Ikbal Leblebicioglu Kurtulus: conceptualization, reviewing, investigation, validation, data curation; Dervis Karaboga: conceptualization, methodology, reviewing, supervision, validation, writing—review and editing; Kerem Kilic: conceptualization, reviewing, investigation, validation, data curation; Alper Basturk: conceptualization, methodology, reviewing, supervision, validation, writing—review and editing; Bahriye Akay: conceptualization, methodology, reviewing, supervision, validation, writing—review and editing; Ozkan Ufuk Nalbantoglu: conceptualization, methodology, reviewing, validation, writing—review and editing; Ozden Melis Durmaz Yilmaz: conceptualization, reviewing, investigation, validation, data curation; Mustafa Ayata: conceptualization, reviewing, investigation, validation, data curation; Serkan Yilmaz: conceptualization, reviewing, investigation, validation, data curation; Ishak Pacal: conceptualization, methodology, software, investigation, writing—review and editing.

Funding This work was supported by the Scientific and Technological Research Council Of Turkey (TUBITAK) (Grant Numbers: 121E068).

Data Availability Data is available on request from the authors.

Declarations

Informed Consent There is no any patient data used in this research.

Competing Interest The authors declare no competing interests.

Human and Animal Rights The authors did not perform any experiments with animals for conducting this research.

References

- H.W. Elani, J.R. Starr, J.D. Da Silva, G.O. Gallucci, Trends in Dental Implant Use in the U.S., 1999–2016, and Projections to 2026, *J Dent Res* 97 (2018) 1424. <https://doi.org/10.1177/0022034518792567>.
- M.A. Saghiri, P. Freag, A. Fakhrzadeh, A.M. Saghiri, J. Eid, Current technology for identifying dental implants: a narrative review, *Bull Natl Res Cent* 45 (2021) 7. <https://doi.org/10.1186/s42269-020-00471-0>.
- T.G.T.M. T Takahashi K Nozaki, Identification of dental implants using deep learning—pilot study, (2020).
- S.-N.J. Jae-Hong Lee, Efficacy of deep convolutional neural network algorithm for the identification and classification of dental implant systems, using panoramic and periapical radiographs: A pilot study., 99 (2020).
- S.-N.J.J.-H. Lee, Efficacy of deep convolutional neural network algorithm for the identification and classification of dental implant systems, using panoramic and periapical radiographs: A pilot study., 99 (2020).
- H.J. Kong, Classification of dental implant systems using cloud-based deep learning algorithm: an experimental study, *Journal of Yeungnam Medical Science* 40 (2023) S29–S36. <https://doi.org/10.12701/JYMS.2023.00465>.
- S. Sukegawa, K. Yoshii, T. Hara, K. Yamashita, K. Nakano, N. Yamamoto, H. Nagatsuka, Y. Furuki, Deep Neural Networks for Dental Implant System Classification, *Biomolecules* 2020, Vol. 10, Page 984 10 (2020) 984. <https://doi.org/10.3390/BIOM10070984>.
- M. Alaftekin, I. Pacal, K. Cicek, Real-time sign language recognition based on YOLO algorithm, *Neural Comput Appl* (2024). <https://doi.org/10.1007/s00521-024-09503-6>.
- J.H. Lee, Y.T. Kim, J. Bin Lee, S.N. Jeong, A Performance Comparison between Automated Deep Learning and Dental Professionals in Classification of Dental Implant Systems from Dental Imaging: A Multi-Center Study, *Diagnostics* 2020, Vol. 10, Page 910 10 (2020) 910. <https://doi.org/10.3390/DIAGNOSTICS10110910>.
- A. Jokstad, U. Braegger, J.B. Brunski, A.B. Carr, I. Naert, A. Wennerberg, Quality of dental implants, *Int Dent J* 53 (2003) 409–443. <https://doi.org/10.1111/J.1875-595X.2003.TB00918.X>.
- I. Pacal, Enhancing crop productivity and sustainability through disease identification in maize leaves: Exploiting a large dataset with an advanced vision transformer model, *Expert Syst Appl* 238 (2024) 122099. <https://doi.org/10.1016/J.ESWA.2023.122099>.
- I. Pacal, MaxCerVixT: A novel lightweight vision transformer-based Approach for precise cervical cancer detection, *Knowl Based Syst* 289 (2024) 111482. <https://doi.org/10.1016/j.knosys.2024.111482>.
- A. Karaman, D. Karaboga, I. Pacal, B. Akay, A. Basturk, U. Nalbantoglu, S. Coskun, O. Sahin, Hyper-parameter optimization of deep learning architectures using artificial bee colony (ABC) algorithm for high performance real-time automatic colorectal cancer (CRC) polyp detection, *Applied Intelligence* 53 (2023) 15603–15620. <https://doi.org/10.1007/s10489-022-04299-1>.
- C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015. <https://doi.org/10.1109/CVPR.2015.7298594>.
- M. Shafiq, Z. Gu, Deep Residual Learning for Image Recognition: A Survey, *Applied Sciences (Switzerland)* 12 (2022). <https://doi.org/10.3390/app12188972>.
- D. Deporter, A.A. Khoshkhounejad, N. Khoshkhounejad, M. Ketabi, A new classification of peri implant gaps based on gap location (A case series of 210 immediate implants), *Dent Res J (Isfahan)* 18 (2021) 29. <https://doi.org/10.4103/1735-3327.313124>.
- I. Pacal, D. Karaboga, A robust real-time deep learning based automatic polyp detection system, *Comput Biol Med* 134 (2021) 104519. <https://doi.org/10.1016/J.COMPBIOMED.2021.104519>.
- I. Pacal, S. Kılıncarslan, Deep learning-based approaches for robust classification of cervical cancer, *Neural Comput Appl* 35 (2023) 18813–18828. <https://doi.org/10.1007/s00521-023-08757-W/METRICS>.
- A. Esteva, A. Robicquet, B. Ramsundar, V. Kuleshov, M. DePristo, K. Chou, C. Cui, G. Corrado, S. Thrun, J. Dean, A guide to deep learning in healthcare, *Nat Med* 25 (2019) 24–29. <https://doi.org/10.1038/s41591-018-0316-z>.
- I. Pacal, A. Karaman, D. Karaboga, B. Akay, A. Basturk, U. Nalbantoglu, S. Coskun, An efficient real-time colonic polyp detection with YOLO algorithms trained by using negative samples and large datasets, *Comput Biol Med* 141 (2022) 105031. <https://doi.org/10.1016/J.COMPBIOMED.2021.105031>.
- S. Shamshirband, M. Fathi, A. Dehzangi, A.T. Chronopoulos, H. Alinejad-Rokny, A review on deep learning approaches in healthcare systems: Taxonomies, challenges, and open issues, *J Biomed Inform* 113 (2021) 103627. <https://doi.org/10.1016/j.jbi.2020.103627>.
- M. Alhanjouri, M. A. H. Lubbad, M. Z. Alkurdi, Robust Speaker Identification using Denoised Wave Atom and GMM, *Int J Comput Appl* 67 (2013) 17–23. <https://doi.org/10.5120/11391-6687>.
- A. Karaman, I. Pacal, A. Basturk, B. Akay, U. Nalbantoglu, S. Coskun, O. Sahin, D. Karaboga, Robust real-time polyp detection system design based on YOLO algorithms by optimizing activation functions and hyper-parameters with artificial bee colony (ABC), *Expert Syst Appl* 221 (2023) 119741. <https://doi.org/10.1016/J.ESWA.2023.119741>.

24. M. Lubbad, D. Karaboga, A. Basturk, B. Akay, U. Nalbantoglu, I. Pacal, Machine learning applications in detection and diagnosis of urology cancers: a systematic literature review, *Neural Comput Appl* (2024) 1–25. <https://doi.org/10.1007/S00521-023-09375-2/METRICS>.
25. I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative Adversarial Networks, *Sci Robot* 3 (2014) 2672–2680. <https://arxiv.org/abs/1406.2661v1> (accessed February 6, 2024).
26. K. O’Shea, R. Nash, An Introduction to Convolutional Neural Networks, *Int J Res Appl Sci Eng Technol* 10 (2015) 943–947. <https://doi.org/10.22214/ijraset.2022.47789>.
27. Z. Li, F. Liu, W. Yang, S. Peng, J. Zhou, A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects, *IEEE Trans Neural Netw Learn Syst* 33 (2022) 6999–7019. <https://doi.org/10.1109/TNNLS.2021.3084827>.
28. Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, *Proceedings of the IEEE* 86 (1998). <https://doi.org/10.1109/5.726791>.
29. A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, *Commun ACM* 60 (2017). <https://doi.org/10.1145/3065386>.
30. I. Pacal, D. Karaboga, A. Basturk, B. Akay, U. Nalbantoglu, A comprehensive review of deep learning in colon cancer, *Comput Biol Med* 126 (2020) 104003. <https://doi.org/10.1016/J.COMPBIOMED.2020.104003>.
31. K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, *IEEE Trans Pattern Anal Mach Intell* 42 (2020). <https://doi.org/10.1109/TPAMI.2018.2844175>.
32. M. Tan, Q. V Le, EfficientNet: Rethinking model scaling for convolutional neural networks, in: 36th International Conference on Machine Learning, ICML 2019, 2019.
33. A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, (2017).
34. J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, T. Chen, Recent advances in convolutional neural networks, *Pattern Recognit* 77 (2018) 354–377. <https://doi.org/10.1016/J.PATCOG.2017.10.013>.
35. I. Leblebicioglu, M. Lubbad, O. M. D. Yilmaz, K. Kilic, D. Karaboga, A. Basturk, ... & I. Pacal. A robust deep learning model for the classification of dental implant brands, *Journal of Stomatology, Oral and Maxillofacial Surgery* (2024) 101818.
36. S. Albawi, T.A. Mohammed, S. Al-Zawi, Understanding of a convolutional neural network, *Proceedings of 2017 International Conference on Engineering and Technology, ICET 2017 2018-January* (2017) 1–6. <https://doi.org/10.1109/ICENGTECHNOL.2017.8308186>.
37. R. Yamashita, M. Nishio, R.K.G. Do, K. Togashi, Convolutional neural networks: an overview and application in radiology, *Insights Imaging* 9 (2018) 611–629. <https://doi.org/10.1007/S13244-018-0639-9/FIGURES/15>.
38. J. Heaton, Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning, *Genet Program Evolvable Mach* 19 (2018) 305–307. <https://doi.org/10.1007/s10710-017-9314-z>.
39. Y. Lecun, Y. Bengio, G. Hinton, Deep learning, *Nature* 521 (2015) 436–444. <https://doi.org/10.1038/NATURE14539>.
40. M.E. Klontzas, G. Kalarakis, E. Koltsakis, T. Papatomas, A.H. Karantanas, A. Tzortzakakis, Convolutional neural networks for the differentiation between benign and malignant renal tumors with a multicenter international computed tomography dataset, *Insights Imaging* 15 (2024) 1–11. <https://doi.org/10.1186/S13244-023-01601-8/FIGURES/5>.
41. H. Habibi Aghdam, E. Jahani Heravi, Guide to Convolutional Neural Networks, Guide to Convolutional Neural Networks (2017). <https://doi.org/10.1007/978-3-319-57550-6>.
42. N. Kalchbrenner, E. Grefenstette, P. Blunsom, A Convolutional Neural Network for Modelling Sentences, 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014 - Proceedings of the Conference 1 (2014) 655–665. <https://doi.org/10.3115/v1/p14-1062>.
43. N. Ketkar, J. Moolayil, Convolutional Neural Networks, *Deep Learning with Python* (2021) 197–242. https://doi.org/10.1007/978-1-4842-5364-9_6.
44. N. Ketkar, J. Moolayil, Deep learning with python: Learn Best Practices of Deep Learning Models with PyTorch, *Deep Learning with Python: Learn Best Practices of Deep Learning Models with PyTorch* (2021) 1–306. <https://doi.org/10.1007/978-1-4842-5364-9>.
45. P. Ramachandran, B. Zoph, Q. V Le Google Brain, Searching for Activation Functions, 6th International Conference on Learning Representations, ICLR 2018 - Workshop Track Proceedings (2017). <https://arxiv.org/abs/1710.05941v2> (accessed February 15, 2024).
46. Z. Yang, Z. Yang, Activation Function: Cell Recognition Based on YoLov5s/m, *Journal of Computer and Communications* 9 (2021) 1–16. <https://doi.org/10.4236/JCC.2021.912001>.
47. M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014. https://doi.org/10.1007/978-3-319-10590-1_53.
48. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, L. Fei-Fei, ImageNet Large Scale Visual Recognition Challenge, *Int J Comput Vis* 115 (2015). <https://doi.org/10.1007/s11263-015-0816-y>.
49. K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, 2015.
50. G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, 2017*. <https://doi.org/10.1109/CVPR.2017.243>.
51. S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, *IEEE Trans Pattern Anal Mach Intell* 39 (2017). <https://doi.org/10.1109/TPAMI.2016.2577031>.
52. J.-S.S.Y.-H.J.B.-H.C.J.J.H.J.-E.K.N.-E. Nam, Transfer Learning via Deep Neural Networks for Implant Fixture System Classification Using Periapical Radiographs., 9 (2020).
53. J. Xu, Y. Pan, X. Pan, S. Hoi, Z. Yi, Z. Xu, RegNet: Self-Regulated Network for Image Classification, *IEEE Trans Neural Netw Learn Syst* 34 (2023). <https://doi.org/10.1109/TNNLS.2022.3158966>.
54. Z. Liu, H. Mao, C.Y. Wu, C. Feichtenhofer, T. Darrell, S. Xie, A ConvNet for the 2020s, in: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2022*. <https://doi.org/10.1109/CVPR52688.2022.01167>.
55. Y. Zhang, J. Wang, J.M. Gorriz, S. Wang, Deep Learning and Vision Transformer for Medical Image Analysis, *J Imaging* 9 (2023) 147. <https://doi.org/10.3390/JIMAGING9070147>.
56. Y.E. Almalki, M. Zaffar, M. Irfan, M.A. Abbas, M. Khalid, K.S. Quraishi, T. Ali, F. Alshehri, S.K. Alduraibi, A.A. Asiri, M.A.A. Basha, A. Alduraibi, M.K. Saeed, S. Rahman, A Novel-based Swin Transfer Based Diagnosis of COVID-19 Patients, *Intelligent Automation & Soft Computing* 35 (2023) 163–180. <https://doi.org/10.32604/IASC.2023.025580>.

57. Y. Li, S. Rao, J.R.A. Solares, A. Hassaine, R. Ramakrishnan, D. Canoy, Y. Zhu, K. Rahimi, G. Salimi-Khorshidi, BEHRT: Transformer for Electronic Health Records, *Scientific Reports* 2020 10:1 10 (2020) 1–12. <https://doi.org/10.1038/s41598-020-62922-y>.
58. T.G.T.M.T.T.K. Nozaki, Identification of dental implants using deep learning—pilot study, (2020).
59. S. Sharma, R. Mehra, Conventional Machine Learning and Deep Learning Approach for Multi-Classification of Breast Cancer Histopathology Images—a Comparative Insight, *J Digit Imaging* 33 (2020). <https://doi.org/10.1007/s10278-019-00307-y>.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.