



## **Dental Implants Classifier**

### **Presented by:**

Mariem Abdou (202000595)

Alaa Hatem (202001170)

Login Darwish (19105985)

Zahraa Omar Elzeiry (18102297)

Shahed Essam Abushaala (18200129)

### **Presented to:**

Dr. Ghada Khoriba

Eng. Amira Tarek Mahmoud

(CSCI 417-3)

# Table of Contents

<b>1 Abstract.....</b>	<b>3</b>
<b>2 Introduction.....</b>	<b>3</b>
<b>3 Related Work .....</b>	<b>4</b>
<b>4 Methodology .....</b>	<b>5</b>
<b>4.1 Dataset.....</b>	<b>5</b>
<b>4.2 Preprocessing.....</b>	<b>6</b>
<b>4.3 Models Architecture .....</b>	<b>7</b>
<b>4.3.1 VGG16 .....</b>	<b>7</b>
<b>4.3.2 ResNet50 .....</b>	<b>8</b>
<b>4.3.3 AlexNet.....</b>	<b>8</b>
<b>4.3.4 GoogleLeNet .....</b>	<b>8</b>
<b>4.3.5 Ensemble model .....</b>	<b>9</b>
<b>4.4 Models Training.....</b>	<b>10</b>
<b>5 Results .....</b>	<b>10</b>
<b>6 Discussion.....</b>	<b>11</b>
<b>7 Conclusion .....</b>	<b>12</b>
<b>8 References.....</b>	<b>13</b>

## **1 Abstract**

The ability to fully chew again is one of an implant's main benefits. Most patients are unable to distinguish between their implant tooth and their natural teeth. They are able to clean their teeth and floss properly while using it, and they can eat normally. However, when an implant fracture occurs, the patient will need repairs. Moreover, since there are many types of dental implants, it is difficult for the dentist to determine the required type. In this study, we employed deep convolutional neural networks (CNNs) to classify and clarify the accuracy of several dental implant brands from X-ray PNG, JPG, and JPEG images. We tried to classify the type of dental implant using different models Alexnet, VGG16 with and without transfer learning, Resnet 50, GoogleLeNet, and Ensemble model. We got different accuracies in our models. Alexnet got 20%, VGG16 without transfer learning and with transfer learning got 2% and 54% respectively, Resnet50 got 6%, and GoodleLeNet got 10%. Meanwhile, the ensemble model got an accuracy of 4.47%. Therefore, it identified that the best model is VGG16.

## **2 Introduction**

The use of dental implants to support tooth replacement has a long and complex history because tooth loss is highly prevalent and can arise from trauma, disease, and other factors. According to data from the American Association of Oral and Maxillofacial Surgeons, 69% of adults between the ages of 35 and 44 have lost at least one permanent tooth due to decay, gum disease, an unsuccessful root canal, or an accident. In addition, 26% of individuals had lost all of their permanent teeth by the age of 74. In light of this, it can be concluded that between 100,000 and 300,000 dental implants are used annually [1].

Dental implants are tiny posts that are affixed to your jaw and serve as a substitute for a tooth's root. With the aid of an abutment, they bond to fresh replacement teeth. Titanium makes up the majority of dental implants. Dental implants come in a variety of varieties, as do the procedures [2]. In order to replace a lost or broken crown of a dental implant, a dentist has to first virtually identify the type of implant used by the patient from x-ray images. As there are more than 50 different types of dental implants that are very similar in shape, it is difficult for a dentist to identify the correct type of implant.

Therefore, deep learning systems use artificial intelligence machine learning methods that let computers learn tasks based on neural networks that are similar to those performed by humans;

as a result, they mirror the neural circuitry in the human brain. When using medical images for categorization and diagnosis, deep learning with CNNs is particularly helpful.

### 3 Related Work

Due to the enormous number of different implant models, identifying dental implants in X-ray pictures can be difficult. To recognize and differentiate between the many dental implant kinds available on the market, a certain level of skill is necessary. The precise classification of an implant type is critical in selecting an appropriate replacement when the old abutment and/or artificial tooth has been lost or damaged. Various dental implant identification systems have been studied. Table I below summarizes the recent literature approaches for implant classification.

TABLE II: Literature review

Reference	Year	Dataset	Preprocessing	Model used	Accuracy
[3]	2020	Kagawa Prefectural Central Hospital (HOPE Dr ABLE-GX, FUJITSU Co., Tokyo, Japan)	- manually cropped - they used Photoshop to manipulate the images so that all dental implant fixtures would fit	- CNN - VGG16-transfer - VGG16-fine tuning - VGG19-transfer - VGG19-fine tuning	<ul style="list-style-type: none"> <li>• 0.860</li> <li>• 0.899</li> <li>• 0.935</li> <li>• 0.880</li> <li>• 0.927</li> </ul>
[4]	2021	7079 Dental panoramic radiographs from Kagawa Prefectural Central Hospital	- manually cropped - resized to $128 \times 128$ pixels	ResNet18 ResNet34 ResNet50 ResNet101 ResNet152	ACC -- F1 score 0.9942 -- 0.9934 0.9951 -- 0.9944 0.9960 -- 0.9948 0.9964 -- 0.9959 0.9955 -- 0.9950
[5]	2020	X-ray images of 801 patients (aged 19–84 years) from Yonsei University Dental Hospital	- cropped manually - data augmentation (rotated from $-20^\circ$ to $20^\circ$ , randomly horizontally rotated, translated horizontally and vertically from $-30$ to $30$ pixels, and scaled horizontally and vertically from $0.7$ to $1.2$ .)	SqueezeNet, GoogLeNet, ResNet-18, MobileNet-v2, and ResNet-50 Pertained model = ImageNet model	ACC -- F1 score 0.96 -- 0.96 0.93 -- 0.93 0.98 -- 0.98 0.97 -- 0.96 0.98 -- 0.98
[6]	2020	Osaka University Graduate School of Dentistry. Dental Hospital	-cropping  - enhancement and removing noise  - resized to $416 \times 416$ pixels	-TensorFlow  -Keras deep learning	- 0.50 to 0.82 - 0.51 to 0.85 - 0.71 to 0.72

[7]	2020	1206 digital radiographs from the databases of 5 practitioners in private practice	<ul style="list-style-type: none"> <li>- Manually cropped</li> <li>- Convert to gray scale</li> <li>- Rotated vertically</li> <li>- Resized to 90x200 px</li> <li>- Data augmentation (horizontal rotation, angulation, tomo, brightness, contrast, blur, sharpness)</li> </ul>	27-layer CNN	ACC = 93.8
[8]	2020	Department of Periodontology, Daejeon Dental Hospital	<ul style="list-style-type: none"> <li>-cropped and labeled by 3 periodontology residents.</li> <li>-removing noise, haziness, and distortion.</li> <li>- remaining images and edit the contrast and brightness using global contrast normalization and zero phase whitening.</li> </ul>	<ul style="list-style-type: none"> <li>- CNN algorithm</li> <li>-VGG-19</li> <li>-Inception-v3</li> <li>-ResNet-50</li> <li>- GoogLeNet</li> <li>Inception-v3</li> </ul>	0.963–0.978 0.913–0.935 0.942–0.967 0.871–0.909 0.969–0.987
[9]	2022	altech 256, PASCAL and Imagenet	<ul style="list-style-type: none"> <li>- cropping images</li> <li>- normalization X-ray images</li> <li>- applied thresholding to the image</li> <li>- ROI-masking</li> </ul>	<ul style="list-style-type: none"> <li>- VGG networks</li> <li>-CNN</li> <li>- GoogLeNet</li> <li>- YOLOv3</li> </ul>	-ACC = 94.0%
[10]	2022	Kagawa Prefectural Central Hospital Picture Archiving and Communication Systems (PACS) (HOPE Dr ABLE-GX, FUJITSU Co., Tokyo, Japan)	<ul style="list-style-type: none"> <li>- All digital image data were converted to a tagged image file format (TIFF) (2964 × 1464, 2804 × 1450, 2694 × 1450, or 2776 × 1450 pixels)</li> <li>- Manually cropped</li> </ul>	<ul style="list-style-type: none"> <li>- CNN(ResNet) with ABN (               <ol style="list-style-type: none"> <li>1. ResNet18</li> <li>2. ResNet18 with ABN</li> <li>3. ResNet50</li> <li>4. ResNet152</li> <li>5. ResNet152 with ABN)</li> </ol> </li> </ul>	<ol style="list-style-type: none"> <li>1. 0.9486</li> <li>2. 0.9719</li> <li>3. 0.9578</li> <li>4. 0.9511</li> <li>5. 0.9624</li> </ol>

## 4 Methodology

This section will go over the dataset, preprocessing, model design, and training specifications.

### 4.1 Dataset

The dataset consists of X-ray PNG, JPG, and JPEG images. The images were gathered online, particularly from Facebook business pages for various firms. The data files were organized

by each company's name and the X-ray image. Exactly 2363 X-ray images were collected from 60 different companies.

## 4.2 Preprocessing

Our data is 60 types of dental implants (60 different folders), each folder contains several X-ray images, and the total number of x-rays is 2363. First, we filtered our types to only use the folders that contain more than 25 x-ray images, and we neglect the types or folders that contain less than 25 images. Second, we separate the data into 10% testing and 90% training. The validation ratio is 20% from the training data. After that, we convert all the images channel to be RGB, so that we don't have issues or errors if an image with PNG extension has an alpha channel which expresses transparency. So, now all images are in 3 channels.

The most important step in the preprocessing is that we take a composition of transformers to make data augmentation, resize, converting to tensor, and normalization. For sure the data augmentation functions were only used on the training dataset, but other functions were used on both datasets.

In data augmentation we use horizontal and vertical flip, and 30 angle rotation. The resized images were 224 pixels. Then we apply the normalization function on the images in training and testing dataset. We prepared the training and testing data for the model. The training and testing data are stored in the train\_images and test\_images directories, respectively, and are assumed to be organized into subdirectories according to class. We first loaded the training and testing data using the Image folder class from PyTorch's dataset module. The Image Folderclass reads the images from the specified directories and returns a dataset object that can be used to access the images and their labels. The train\_transforms and test\_transforms objects are transformations that are applied to the images as they are loaded. These transformations are used to resize images and perform data augmentation. Then we created a (class names) by accessing the classes attribute of the train\_data object. Next, we created a validation set by randomly selecting a subset of the training data and splitting it into training and validation sets. To do this, the code first generates a list of indices that correspond to the data points in the training set, shuffles the indices, and then splits the indices into training and validation sets according to the desired ratio of training to validation data (specified by the validation\_ratio variable). The resulting training and validation indices are stored in the train\_idx and valid\_idx variables, respectively. These indices can be used

to select the corresponding data points from the training set using a PyTorch SubsetRandomSampler.

We created a pytorch DataLoaders for training, validation, and testing data. The DataLoader is an iterator that provides a batch of data to the model for training or evaluation. First, we created two SubsetRandomSampler objects, train\_sampler and valid\_sampler, which are used to select a subset of the data for training and validation. The subset is specified using the train\_idx and valid\_idx indices, which are lists of integers that specify the indices of the training and the indices of validation to include it in the subset. Finally, we created a dictionary called loaders that maps the string keys 'train', 'valid', and 'test' to the corresponding DataLoader objects. This dictionary can be used to easily access the DataLoaders for different splits of the data.

### **4.3 Models Architecture**

This section will contain the details of the five models chosen for implementation: VGG16, ResNet50, AlexNet, GoogleLeNet, and an ensemble model of two models (ResNet50 and AlexNet).

#### **4.3.1 VGG16**

VGG model is a (Visual Geometry Group) network, a convolutional neural network that was developed by the Visual Geometry Group at the University of Oxford. The model consists of a series of convolutional and max pooling layers, followed by fully connected layers and a final classification layer. The model is trained on a large dataset of images and is used for tasks such as object recognition and image classification. The model has a series of convolutional layers that extract features from the input images, and fully connected layers that map the features to class scores. The model also includes ReLU activation functions and dropout layers to improve generalization and prevent overfitting.

We have edited the classifier of the VGG model to avoid overfitting because we don't have big data that can suit this big model without overfitting occurring. The final layer of the model is a linear layer with 41 output units, which corresponds to the number of classes in the dataset. I just added one more layer to the classifier, so that the output features are equal to the number of types of dental implants used.

### **4.3.2 ResNet50**

Deep residual learning is a neural network architecture introduced by He et al. in 2015 [11]. Deep Residual Learning for Image Recognition has been widely referenced and is widely regarded as one of the most significant articles in the field of computer vision. It is an architecture for a convolutional neural network (CNN) that can accommodate hundreds or even thousands of convolutional layers. Previous CNN designs could only support a few layers, which had a negative impact on performance. There was a vanishing gradient issue as more layers were added. Backpropagation is used to train neural networks, which involves moving down the loss function and finding the weights that minimize it. When there are too many layers, the gradient will soon get saturated and worsen with each additional layer, and repeated multiplications will finally cause it to vanish (become zero) [12].

Skip connections, a novel approach offered by ResNet, solves the vanishing gradient issue. IN the beginning, ResNet stacks several convolutional layers, skips those levels, and reuses the prior layer's activations. Since ResNet makes model deterioration less likely to occur, it can be deepened to very deep layers of over 100 layers [4]. Popular ResNet architectures include ResNet18, ResNet34, ResNet101, and ResNet152 which has 18, 34, 50, 101 and 152 layers respectively. In this study, we will be implementing the ResNet50 model.

### **4.3.3 AlexNet**

AlexNet and LeNet both have very similar designs; however, AlexNet is deeper and has more layers of filters. It has eight layers: five convolutional layers some of them are followed by max-pooling layers, three fully connected hidden layers one of them is an output layer [13]. The five convolution layers and the first two fully connected layers use the ReLu function as an activation function. The output layer uses Softmax as an activation function. There are two dropout layers, one after the convolution layers, and the other after the first fully connected layer [14].

### **4.3.4 GoogleLeNet**

GoogLeNet is a 22-layer deep convolutional neural network that's a variant of the Inception Network, a Deep Convolutional Neural Network developed by researchers at Google. The early development of very effective solutions to fundamental computer vision problems was facilitated by the introduction of CNN, bigger datasets, effective processing resources, and intuitive CNN architectures. Researchers found that adding layers and units to a network



significantly improved performance. However, adding more layers to build larger networks has a price. Large networks can overfit and experience either an inflating gradient problem or a vanishing gradient problem. Most of the issues that huge networks had were resolved by the GoogleLeNet design, primarily through the use of the Inception module. The Inception module is a neural network architecture that leverages feature detection at different scales through convolutions with different filters and reduced the computational cost of training an extensive network through dimensional reduction. The GoogLeNet architecture consists of nine inception modules. Notably, there are two max-pooling layers between some inception modules. The purpose of these max-pooling layers is to downsample the input as it's fed forward through the network [15].

This is achieved through the reduction of the height and width of the input data. Another efficient way to lower the computational strain on the network is to reduce the input size between the inception modules. The input height and width are lowered to 1x1, and the average pooling layer averages all the feature maps created by the previous inception module. In order to avoid overfitting, the network during training, the dropout layer is a regularization approach. There are 1000 hidden units in the linear layer. The SoftMax layer, the last layer, is used to calculate the probability distribution of a set of integers included in an input vector. The SoftMax function is an activation function. A vector with a collection of values that reflects the likelihood of an event or class occurring is the result of a SoftMax activation function. The vector's values sum up to one when totaled. In conclusion, the categorization tasks were successfully completed using the GoogleNet architecture. Any deep learning practitioner who wants to comprehend the advancement of deep conv networks within the deep learning community must first grasp the GoogLeNet architecture. It is possible to learn a great deal by going back and reviewing research projects from years past [15].

#### **4.3.5 Ensemble model**

Ensemble techniques are often regarded as the cutting-edge answer to many machine learning problems. By training many models and aggregating their predictions, such strategies increase the predictive performance of a single model. The primary concept of ensemble learning is that by merging many models, the fault of a single model is likely to be compensated for by other models, resulting in the ensemble's total prediction performance being better than that of a

single model [16]. Therefore, we would like to build an ensemble model of the two models mentioned previously (ResNet50 and AlexNet).

#### 4.4 Models Training

The dataset was randomly divided into the training, validation, and testing sets with the ration of 6:2:2 respectively. All models were trained on Google Collab. The ResNet50, AlexNet, and GoogleNet were built using the Python TensorFlow library; however, the VGG16 model was built using the Python PyTorch library. All of the models analyzed a maximum of 50 epochs and minibatch size 64. We used the early stop method to terminate data training to prevent overfitting if the validation error did not update 5 times in a row. The model with the greatest performance on a validation dataset for each pre-trained network was chosen. If the model performed the same for numerous validation patience, the model with the least validation patience was chosen.

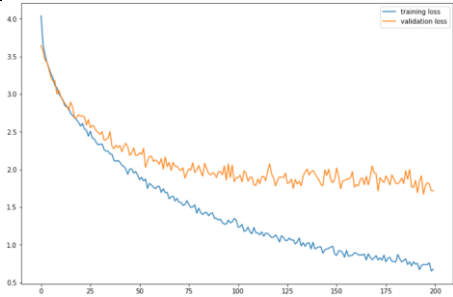
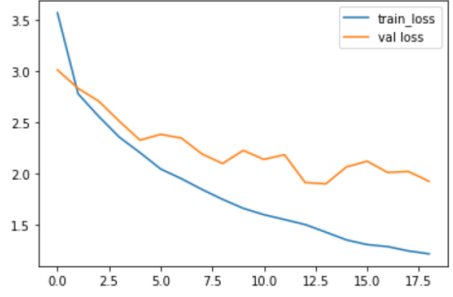
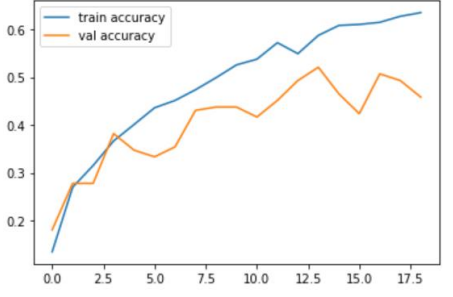
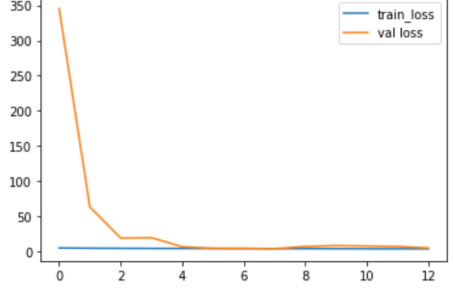
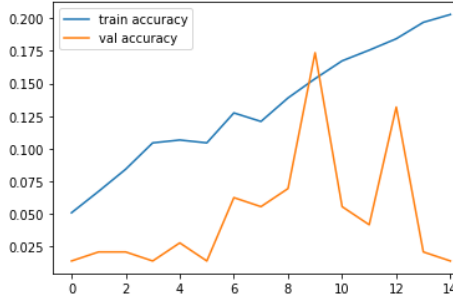
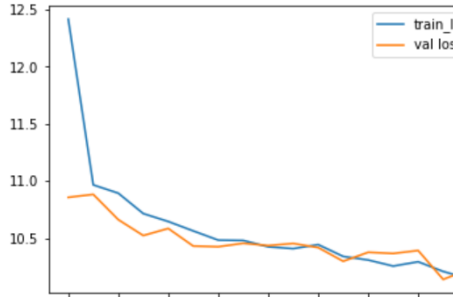
#### 5 Results

The cross-entropy loss function of the chosen training image dataset was utilized to train the CNN models in this work. Table II below displays the dental implant classification results for all five models. Table III shows the loss and accuracy plots for the four models trained.

TABLE II: Models Accuracy

Model	Training Acc	Validation Acc	Testing Accuracy
<b>VGG16</b>	-	36.456	54
<b>ResNet50</b>	6.351	4.583	5.472
<b>AlexNet</b>	20	1	11
<b>GoogleLeNet</b>	10.61	11.11	5.97
<b>Ensemble model</b>	-	-	04.47

TABLE III: Models Loss and Accuracy plots

Model	Loss plot	Accuracy plot
<b>VGG16</b>		-
<b>ResNet50</b>		
<b>AlexNet</b>		
<b>GoogleLeNet</b>		

## 6 Discussion

The results show that VGG16 model had the best performance with a testing accuracy of 54%. The next best performer was the AlexNet model which had a testing accuracy of 11%. On the other hand, there was no clear difference between the ResNet50 model and GoogleLeNet

model accuracies. The plots reveal that the AlexNet model as well as the ResNet50 model started to overfit around the 7<sup>th</sup> epoch. The VGG16 model relatively high performance is likely since the model was pre-trained on the Xavier dataset. The usage of an average ensemble model can be used to explain the ensemble model's weak performance. As we convert to a weighted aggregate strategy of all models, the performance will improve.

## **7 Conclusion**

In our study, we demonstrated that the deep CNNs analyzed could classify the dental implant systems retrieved from X-ray pictures with a variety of accuracy levels. We used different models with different accuracies to classify the dental implants. The best CNN model with the highest accuracy was VGG16.

## 8 References

- [1] L. Gaviria, J. P. Salcido, T. Guda, and J. L. Ong, "Current trends in dental implants," *J Korean Assoc Oral Maxillofac Surg*, vol. 40, no. 2, p. 50, 2014, doi: 10.5125/jkaoms.2014.40.2.50.
- [2] K. Robinson, "Types of Dental Implants," Apr. 26, 2022.
- [3] S. Sukegawa *et al.*, "Deep neural networks for dental implant system classification," *Biomolecules*, vol. 10, no. 7, pp. 1–13, Jul. 2020, doi: 10.3390/biom10070984.
- [4] S. Sukegawa *et al.*, "Multi-task deep learning model for classification of dental implant brand and treatment stage using dental panoramic radiograph images," *Biomolecules*, vol. 11, no. 6, Jun. 2021, doi: 10.3390/biom11060815.
- [5] J. E. Kim, N. E. Nam, J. S. Shim, Y. H. Jung, B. H. Cho, and J. J. Hwang, "Transfer learning via deep neural networks for implant fixture system classification using periapical radiographs," *J Clin Med*, vol. 9, no. 4, Apr. 2020, doi: 10.3390/jcm9041117.
- [6] T. Takahashi, K. Nozaki, T. Gonda, T. Mameno, M. Wada, and K. Ikebe, "Identification of dental implants using deep learning—pilot study," *Int J Implant Dent*, vol. 6, no. 1, Dec. 2020, doi: 10.1186/s40729-020-00250-6.
- [7] M. Hadj Saïd, M.-K. le Roux, J.-H. Catherine, and R. Lan, "Development of an Artificial Intelligence Model to Identify a Dental Implant from a Radiograph," *Int J Oral Maxillofac Implants*, vol. 35, no. 6, pp. 1077–1082, Nov. 2020, doi: 10.11607/jomi.8060.
- [8] J. H. Lee and S. N. Jeong, "Efficacy of deep convolutional neural network algorithm for the identification and classification of dental implant systems, using panoramic and periapical radiographs: A pilot study," *Medicine*, vol. 99, no. 26, p. e20787, Jun. 2020, doi: 10.1097/MD.00000000000020787.
- [9] A. Kohlakala, J. Coetzer, J. Bertels, and D. Vandermeulen, "Deep learning-based dental implant recognition using synthetic X-ray images," *Med Biol Eng Comput*, 2022, doi: 10.1007/s11517-022-02642-9.
- [10] S. Sukegawa *et al.*, "Is attention branch network effective in classifying dental implants from panoramic radiograph images by deep learning?," *PLoS One*, vol. 17, no. 7 July, Jul. 2022, doi: 10.1371/journal.pone.0269016.
- [11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition." [Online]. Available: <http://image-net.org/challenges/LSVRC/2015/>
- [12] M. Shafiq and Z. Gu, "Deep Residual Learning for Image Recognition: A Survey," *Applied Sciences*, vol. 12, no. 18, p. 8972, Sep. 2022, doi: 10.3390/app12188972.
- [13] K. Le, "Alexnet and image classification," Mar. 10, 2021.
- [14] S. Saxena, "Introduction to The Architecture of Alexnet," Mar. 19, 2021.
- [15] R. Alake, "Deep Learning: GoogLeNet Explained," Dec. 23, 2020.
- [16] O. Sagi and L. Rokach, "Ensemble learning: A survey," *WIREs Data Mining and Knowledge Discovery*, vol. 8, no. 4, Jul. 2018, doi: 10.1002/widm.1249.