Project Report

Object and disease detection using X-ray photos

ImplantDetect: Automated Dental Implant Detection using Linear Algebra Techniques (ImplantDetect-LA)

Liliana Hotsko, Yurii Kharabara

Introduction

The teeth are the most challenging material to work with in the human body. Existing methods for detecting teeth problems are characterized by low efficiency, the complexity of the experiential operation, and a higher level of user intervention. Older oral disease detection approaches were manual, time-consuming, and required a dentist to examine and evaluate the disease.

The research area of the project focuses on dental imaging analysis and the application of computer vision and linear algebra techniques to automate the detection of dental implants in X-ray scans. The problem this project aims to solve is to develop an automated system that can efficiently assess the presence and quantity of dental implants (and others in future) in patients' X-ray images, ultimately improving dental care efficiency.

Background and Related Work

Dental implant detection is an image analysis task that has been the subject of several studies. Various approaches have been proposed for this task, including classical image processing techniques, linear algebra-based methods, and deep learning techniques.

Classical image processing techniques involve the use of various filters, edge detection methods, and segmentation algorithms to identify dental implants in X-ray images. Although these techniques can be effective in certain cases, they are often sensitive to noise and may require manual tuning of parameters for optimal performance.

Deep learning techniques, such as convolutional neural networks (CNNs), have gained popularity in recent years due to their ability to learn hierarchical features from raw data. CNNs have shown great potential in image analysis tasks, including dental implant detection. However, training a deep learning model can be computationally expensive and may require a large amount of labeled data for effective learning.

Linear algebra techniques, which can potentially lead to a computationally efficient solution, such as singular value decomposition (SVD) and principal component analysis (PCA), offer an alternative approach for dental implant detection in X-ray images. These techniques have been successfully applied to various image analysis tasks and can potentially lead to a computationally efficient solution for "dental detection".

Review of the related work and possible approaches to solutions

In the related work [1], the authors mention the use of classical image processing techniques and deep learning techniques, such as CNNs.
The project uses a U-Net model and binary image analysis for automatic semantic segmentation and measurement of total length of teeth in panoramic X-ray images. The dataset used in this project is different from the one used in our.

Both projects aim to improve dental care through automation and use of computer vision techniques. The ImplantDetect-LA project focuses on implant detection(and other things) using linear algebra techniques, while the semantic segmentation project focuses on tooth segmentation using deep learning techniques.

Possible approaches

Singular Value Decomposition (SVD) is a method that factorizes a matrix into three matrices: the left singular vectors, the singular values, and the right singular vectors. Mathematically, for a given matrix A, SVD can be represented as:

$A = U\Sigma VT$

where U contains the left singular vectors, Σ is a matrix containing the singular values on its diagonal (and other entities equals to 0), and V^T contains the right singular vectors.

SVD is particularly useful for dimensionality reduction and feature extraction. In this project, SVD could be applied to the dental X-ray images to extract the most significant features that help in implant detection.

Principal Component Analysis (PCA) is an unsupervised linear algebra technique that transforms data into a new coordinate system. It reduces the dimensionality of the data by retaining only the most important components that capture the highest variance. PCA generally tries to find the lower-dimensional surface to project the high-dimensional data.

Some common terms used in PCA algorithm:

Dimensionality: It is the number of features or variables present in the given dataset. More easily, it is the number of columns present in the dataset.

Correlation: It signifies how strongly two variables are related to each other. Such as if one changes, the other variable also gets changed. The correlation value ranges from -1 to +1.
Here, -1 occurs if variables are inversely proportional to each other, and +1 indicates that variables are directly proportional to each other.

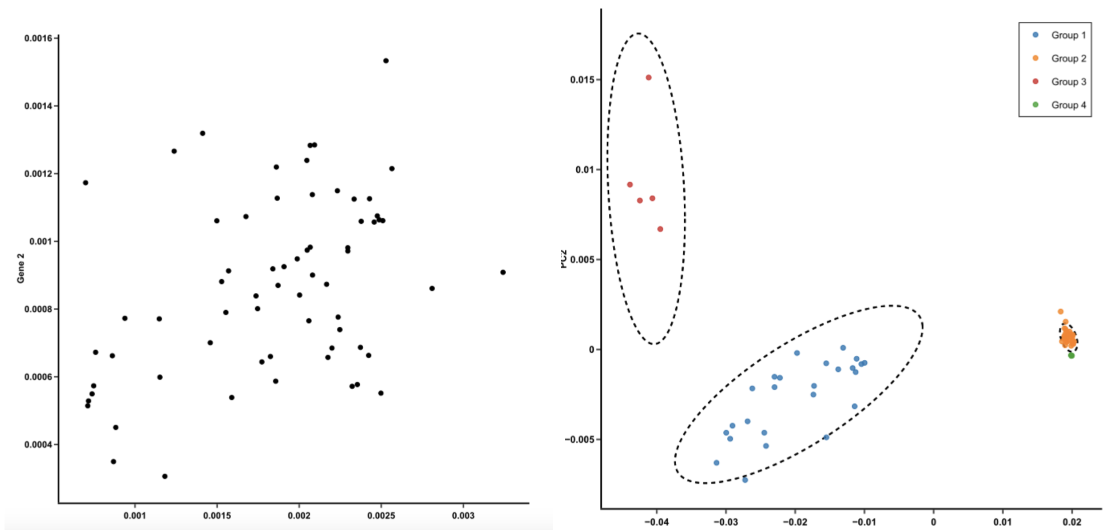
Orthogonal: It defines that variables are not correlated to each other, and hence the correlation between the pair of variables is zero.

Eigenvectors: If there is a square matrix M, and a nonzero vector v is given. Then v will be an eigenvector if Av is the scalar multiple of v.

Covariance Matrix: A matrix containing the covariance between the pair of variables is called the Covariance Matrix.

Description of the chosen algorithm:

It is a statistical process that converts the observations of correlated features into a set of linearly uncorrelated features with the help of orthogonal transformation. These new transformed features are called the Principal Components. It is a technique to draw strong patterns from the given dataset by reducing the variances.



These pictures show how PCA brings out strong patterns from large and complex datasets.

PCA works by considering the variance of each attribute because the high attribute shows the good split between the classes, and hence it reduces the dimensionality. Some real-world applications of PCA are image processing, movie recommendation systems, and optimizing the power allocation in various communication channels. It is a feature extraction technique, so it contains the important variables and drops the least important variable.
PCA is used in this project to further reduce the dimensionality of the image features and retain only the most relevant information for implant detection.

Mathematically, PCA is an algorithm to transform the columns of a dataset into a new set of features called Principal Components.

By doing this, a large chunk of the information across the full dataset is effectively compressed in fewer feature columns. This enables dimensionality reduction and ability to visualize the separation of classes or clusters if any.

The information contained in a column is the amount of variance it contains. The primary objective of Principal Components is to represent the information in the dataset with minimum columns possible.

The transformed new features or the output of PCA are the Principal Components. The number of them is either equal to or less than the original features present in the dataset.
Some properties of these principal components are given below:

- The principal component must be the linear combination of the original features.
- These components are orthogonal, i.e., the correlation between a pair of variables is zero.
- The importance of each component decreases when going to 1 to n, it means the 1 PC has the most importance, and the n PC will have the least importance.

Steps for PCA algorithm:

Getting the dataset
Firstly, we need to take the input dataset and divide it into two subparts X and Y, where X is the training set, and Y is the validation set.

Representing data into a structure
Representing data as a two-dimensional matrix of independent variable X. Here each row corresponds to the data items, and the column corresponds to the Features. The number of columns is the dimensions of the dataset.

Standardizing the data
Such as in a particular column, the features with high variance are more important compared to the features with lower variance. If the importance of features is independent of the variance of the feature, then we will divide each data item in a column with the standard deviation of the column. Here we will name the matrix as Z.

Calculating the Covariance of matrix Z
Take the transposed matrix Z and multiply by Z. The output matrix will be the Covariance matrix of Z.

Calculating the Eigen Values and Eigen Vectors for the resultant covariance matrix Z. Eigenvectors or the covariance matrix are the directions of the axes with high information. And the coefficients of these eigenvectors are defined as the eigenvalues.

Sorting the Eigen Vectors
In this step, take all the eigenvalues and will sort them in decreasing order, which means from largest to smallest. And simultaneously sort the eigenvectors accordingly in matrix P of eigenvalues. The resultant matrix will be named as P*.

Calculating the new features Or Principal Components
Multiply the P* matrix to the Z. In the resultant matrix Z*, each observation is the linear combination of original features. Each column of the Z* matrix is independent of each other.

Remove less or unimportant features from the new dataset.
Keep the relevant or important features in the new dataset only.

<u>Pros and Cons of Chosen Algorithm</u>

● Pros:

Computationally efficient: Linear algebra techniques typically require fewer computational resources compared to deep learning techniques.

Effective in preserving essential features of the data: SVD and PCA can retain important information from the dental X-ray images, making them suitable for feature extraction.

Well-suited for image analysis tasks: Linear algebra techniques have been successfully applied to various image analysis tasks, suggesting their potential for dental implant detection.

● Cons:

Sensitive to noise in the data: Linear algebra techniques can be affected by noise in the data, which may impact their performance.

Limited in their ability to capture non-linear relationships: SVD and PCA are linear techniques and may not be able to capture complex non-linear relationships in the data.

May require additional techniques for further optimization: The performance of linear algebra techniques may need to be improved using other methods, such as ensemble techniques or model fine-tuning.

Implementation Pipeline

The implementation pipeline consists of the following steps:

● Data Preprocessing

Dental X-ray images from the Tufts dentistry database and other sources will be cleaned and normalized, ensuring that they are suitable for analysis. This step involves removing artifacts, correcting image orientation, and resizing images to a standard size.

● Feature Extraction

We will apply SVD and PCA to extract relevant features from the dental X-ray images. These features are crucial for detecting and counting dental implants. The extracted features are then used as input for the subsequent model development.

● Model Development

We are currently creating a model based on the extracted features to detect and count dental implants in the images. The model will be trained using a supervised learning approach, such as logistic regression or support vector machines, and will be fine-tuned to optimize its performance.

● Model Evaluation

We will assess the model's performance using metrics such as accuracy, precision, and recall. This step will help us determine the effectiveness of our chosen techniques in detecting dental implants. The model evaluation will also involve comparing our approach with other existing methods to gauge its relative performance.

● Optimization

Based on the model evaluation, we will fine-tune the model to improve its performance. Thismay include exploring other linear algebra techniques, adjusting model parameters, or incorporating additional features.

Challenges

Some challenges that might hamper our work include the quality and diversity of the dental X-ray images in the dataset, which may affect the model's performance. Additionally, the complexity of dental implant shapes and their overlap with other dental structures could make accurate detection difficult. Overcoming these challenges will require rigorous experimentation and optimization of the model and its parameters.

Due to the lack of dental disease datasets, we used 2 already existing labeled dataset and also we get a big part of images from real dentistry on our own. The size of datasets used after augmentation was almost 1000 images.

Conclusion

In this report, we have provided an in-depth overview of the project's progress, detailing the problem we aim to solve, the related work and possible approaches, and the chosen algorithm with its pros and cons. We have also discussed the theoretical background of the linear algebra methods used, described the implementation pipeline, and outlined our plans for future research.

As we continue to develop and optimize our model, we are confident that our approach using linear algebra techniques will yield an effective automated system for dental implant detection in X-ray scans. This system has the potential to significantly improve the efficiency of dental care by providing dental practitioners with a fast and accurate assessment of patients' implant presence and quantity.

DataSet

GitHubRepository

References

[1] https://github.com/SerdarHelli/Segmentation-of-Teeth-in-Panoramic-X-ray-Image-Using-U-Net/blob/master/README.md
[2] H. C. Burger, J. M. F. Moura, and P. S. R. Diniz, "Using linear algebra for intelligent information retrieval," SIAM Review, vol. 37, no. 4, pp. 573–595, 1995.
[3] L. S. Davis, "A survey of edge detection techniques," Computer Graphics and Image Processing, vol. 4, no. 3, pp. 248–270, 1975.
[4] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," Communications of the ACM, vol. 24, no. 6, pp. 381–395, 1981.
[5] R. O. Duda, P. E. Hart, and D. G. Stork, Pattern Classification. John Wiley & Sons, 2000
[6] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," Biological Cybernetics, vol. 36, no. 4, pp. 193–202, 1980.
[7] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," Neural Computation, vol. 4, no. 1, pp. 1–58, 1992.
[8] http://databookuw.com
[9] https://www.youtube.com/playlist?list=PLMrJAkhIeNNSVjnsviglFoY2nXildDCcv