REPUBLIC OF TURKEY
AKDENIZ UNIVERSITY

AUTOMATIC DIAGNOSIS FROM PANORAMIC DENTAL X-RAYS USING DEEP LEARNING

Kaan KÜÇÜK

INSTITUTE OF NATURAL AND APPLIED SCIENCES

DEPARTMENT OF COMPUTER ENGINEERING

MASTER THESIS

AUGUST 2021

ANTALYA

REPUBLIC OF TURKEY
AKDENIZ UNIVERSITY

AUTOMATIC DIAGNOSIS FROM PANORAMIC DENTAL X-RAYS USING DEEP LEARNING

Kaan KÜÇÜK

INSTITUTE OF NATURAL AND APPLIED SCIENCES

DEPARTMENT OF COMPUTER ENGINEERING

MASTER THESIS

AUGUST 2021

ANTALYA

REPUBLIC OF TURKEY

AKDENIZ UNIVERSITY

INSTITUTE OF NATURAL AND APPLIED SCIENCES

AUTOMATIC DIAGNOSIS FROM PANORAMIC DENTAL X-RAYS USING DEEP LEARNING

Kaan KÜÇÜK

INSTITUTE OF NATURAL AND APPLIED SCIENCES

DEPARTMENT OF COMPUTER ENGINEERING

This thesis accepted by the jury on 09/07/2021

Asst. Prof. Dr. Mustafa Berkay YILMAZ (Supervisor)

Asst. Prof. Dr. Hüseyin Gökhan AKÇAY

Asst. Prof. Dr. Yusuf ÖZÇEVİK

ÖZET

DERİN ÖĞRENMEYİ KULLANARAK PANORAMİK DİŞ RÖNTGENLERİNDEN TEŞHİS KOYMAK

Kaan KÜÇÜK

Yüksek Lisans Tezi, Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Mustafa Berkay YILMAZ

Ağustos 2021; 45

Diş hekimliğinde radyoloji; ağız, diş ve çene bölgesindeki geçmiş dönemde yapılan tedavilerin, hastalıkların ve ileride doğabilecek sorunların tespit edildiği ilk basamaktır. Çekilen panoramik röntgenler sayesinde hastanın mevcut durumu bir bütün olarak ele alınarak bir hekimin incelemeleri sonucunda gerekli tedaviler planlanmaktadır. Fakat artan nüfus, yetersiz hekim sayısı, yetersiz hastaneler ve ekonomik sıkıntıların bir sonucu olarak hastalar ağrılı bir şekilde önce röntgen sırası daha sonra da hekim kontrolü için çok sıra beklemektedir.

Bu tezde, panoramik diş röntgenlerinden hastanın geçmiş dönemde yapılan tedavilerinin tespitinin derin öğrenme metotlarıyla tespiti sunulmuştur. Çalışmanın hekimlerin hastanın geçmiş tanılarını bulmasına hız kazandırması ve gelecekte hastalıkların ve doğabilecek sorunların bulunabilmesi için bir ön hazırlık olması beklenmektedir.

Bu çalışma kapsamında, sağlık alanında yapılan 2 boyutlu veya 3 boyutlu röntgenlerden hastalık tespiti yapan önceki çalışmalar incelenmiştir. Derin öğrenme yöntemi olarak CNN algoritmasının, nesne tespiti içinse verimlilik ve hız faktörleri göz önüne alınınca YOLO algoritmasının yaygın olarak kullanıldığı görülmüştür. Bazı çalışmalarda ise bölge önerimi için RPN ve nesne tespiti için SSD, Faster R-CNN gibi teknikler kullanılmıştır. Bu çalışmada aynı veri seti kullanılarak revize edilmiş YOLO versiyonları ve revize edilmiş Faster R-CNN modeli aynı kapasitedeki bir bilgisayarda denenmiştir. Böylece bu yöntemler arasında başarı oranı ve hız gibi önemli etkenlerin karşılaştırılması da yapılmıştır.

ANAHTAR KELİMELER: Derin öğrenme, Konvolüsyonel Sinir Ağları, Obje Tespiti ve Sınıflandırılması.

JÜRİ: Dr. Öğr. Üyesi Mustafa Berkay YILMAZ

Dr. Öğr. Üyesi Hüseyin Gökhan AKÇAY

Dr. Öğr. Üyesi Yusuf ÖZÇEVİK

ABSTRACT

AUTOMATIC DIAGNOSIS FROM PANORAMIC DENTAL X-RAYS USING DEEP LEARNING

Kaan KÜÇÜK

MSc Thesis in Computer Engineering

Supervisor: Asst. Prof. Dr. Mustafa Berkay YILMAZ

August 2021; 45 pages

Radiology in dentistry is the first step in detecting past treatments, diseases, and problems that may arise in the future in the area of the mouth, teeth, and jaw. Thanks to the panoramic X-rays taken, the patient's current condition is considered as a whole, and the necessary treatments are planned because of the doctor's examinations. But because of the growing population, an insufficient number of doctors, insufficient hospitals, and economic difficulties, patients painfully wait a lot of rows first for an X-ray, and then for a doctor's check-up.

In this thesis, the determination of the patient's past treatments from panoramic dental X-rays by the deep learning method is presented. It is expected that the study will accelerate doctors to find the patient's past diagnoses and be a preliminary preparation for finding diseases and problems that may arise in the future.

As part of this study, previous studies that detected diseases from 2-dimensional or 3-dimensional X-rays conducted in the field of health were examined. It has been observed that the CNN algorithm is widely used as a deep learning method, and the YOLO algorithm is widely used for object detection, considering efficiency and speed factors. In some studies, techniques such as RPN for zone suggestion and SSD for object detection, Faster R-CNN have been used. In this study, the revised YOLO versions and the revised Faster R-CNN model were tested on a computer with the same capacity using the same data set. In this way, important factors such as success rate and speed were compared between these methods.

KEYWORDS: Classification, Convolutional Neural Networks, Deep Learning, Object Detection.

COMMITTEE: Asst. Prof. Dr. Mustafa Berkay YILMAZ

Asst. Prof. Dr. Hüseyin Gökhan AKÇAY

Asst. Prof. Dr. Yusuf ÖZÇEVİK

ACKNOWLEDGEMENTS

I would like to thank my supervisor Asst. Prof. Dr. Mustafa Berkay YILMAZ for devoting his time, share his knowledge and experience. He has always been patient and supportive. I would also like to thank my sister Asst. Prof. Dr. Şirin KÜÇÜK AVCI, who encouraged me to pursue my master's degree at the beginning and always supported me during my education. I would also like to thank my mother and father, whose love and support I always felt by my side.

LIST OF CONTENTS

ÖZET i
ABSTRACT ii
ACKNOWLEDGEMENTS iii
TEXT OF OATH vi
ABBREVIATIONS vii
LIST OF FIGURES viii
LIST OF TABLES ix
1. INTRODUCTION 1
2. LITERATURE REVIEW 3
2.1. Convolutional Neural Networks 3
2.2. Object Detection 4
2.3. Deep Learning in Healthcare 5
2.4. Deep Learning in Dentistry 6
3. MATERIAL AND METHOD 8
3.1. Data Set 8
3.1.1. Data Set Collection 8
3.1.2. Labeling data set 8
3.2. Convolutional Neural Network 10
3.2.1. Feature extraction 11
3.2.2. Padding 12
3.2.3 Pooling and stride 12
3.2.4. Fully-Connected Layer 13
3.3. Faster R-CNN Model 13
3.3.1. Tensorflow object detection API 14
3.3.2. Training Procedure 15
3.4. YOLO Model 15
3.4.1. YOLOv3 15
3.4.2. Setting up Parameters of YOLO 16
4. RESULTS AND DISCUSSIONS 18
4.1. Experiment Details 18
4.2. Experiment Results 20

4.2.1. Precision, Recall, F1 score and Map 20
4.2.2. Loss function 21
4.3. Results of models 21
4.3.1. YOLOv3-tiny model 21
4.3.2. YOLOv3 model 23
4.3.3. YOLOv4-tiny model 24
4.3.4. YOLOv4 model 26
4.3.5. Faster R-CNN model 28
4.4. Comparison of models 29
5. CONCLUSION 30
6. REFERENCES 31

TEXT OF OATH

I declare that this study "Automatic Diagnosis from Panoramic Dental X-Rays Using Deep Learning", which I present as my master thesis, is following the academic rules and ethical conduct. I also declare that I cited and referenced all material and results that are not original to this work.

06/08/2021

Kaan KÜÇÜK

ABBREVIATIONS

Abbreviations:

CNN: Convolutional Neural Networks

YOLO: You Only Look Once

R-CNN: Region-Based CNN

SSD: Single Shot Detector

MLP: Multilayer Perceptron

GPU: Graphics Processing Unit

SVM: Support Vector Machine

mAP: mean Average Precision

LIST OF FIGURES

Figure 1.1. Panoramic Dental Image. 1
Figure 1.2. AI Venn Diagram(Raza & Cinquegrana, 2018). 2
Figure 2.1. LeNet architecture (Lecun et al., 1998)) 3
Figure 2.2. SSD architecture (Liu et al., 2016) 4
Figure 2.3. YOLO architecture (Redmon et al., 2016). 5
Figure 3.1. Distribution of Classes in Data Set 8
Figure 3.2. Txt file with annotation 9
Figure 3.3. LabelIMG 10
Figure 3.4. Complete CNN model 11
Figure 3.5. Convolution Operation(Wang et al., 2019) 11
Figure 3.6. Pooling operation. 12
Figure 3.7. Fully-Connected Layer. 13
Figure 3.8. Faster R-CNN 14
Figure 3.9. Coco Trained Models 14
Figure 3.10. YOLOv3 106 layers. 16
Figure 4.1. Images with data augmentation 19
Figure 4.2. Confusion matrix 20
Figure 4.3. YOLOv3 – tiny loss function 22
Figure 4.4. YOLOv3 – tiny Precision, Recall, F1 score and mAP values 22
Figure 4.5. YOLOv3 – tiny result. 23
Figure 4.6. YOLOv3 loss function 23
Figure 4.7. YOLOv3 Precision, Recall, F1 score and mAP values 24
Figure 4.8. YOLOv3 result. 24
Figure 4.9. YOLOv4 – tiny loss function. 25
Figure 4.10. YOLOv4 – tiny Precision, Recall, F1 score and mAP values. 25
Figure 4.11. YOLOv4 – tiny result. 26
Figure 4.12. YOLOv4 loss function. 26
Figure 4.13. YOLOv4 Precision, Recall, F1 score and mAP values. 27
Figure 4.14. YOLOv4 result. 27
Figure 4.15. Faster R-CNN loss function. 28
Figure 4.16. Faster R-CNN result. 28

LIST OF TABLES

Table 4.1. System Features provided by Google Colab Pro(Anonymous). . . . 18
Table 4.2. Comparison of models. 29

1. INTRODUCTION

The working logic of dental hospitals around the world is as follows: The patient comes to the dental hospital. Whatever the complaint, he is referred to the X-ray clinic first. After the X-ray is taken, it is diagnosed by a doctor in the oral diagnosis clinic. Then the patient is referred to other clinics according to the diagnosis. For example, if he needs filling, he can go to the restorative clinic. But it is thought that this is a huge waste of time for a patient who has pain, especially in largely populated cities.

A dentist should look at the patient's x-ray and take note of previous treatments. This is the first step in understanding whether the patient's complaint is due to past treatments or is a new condition. These procedures are generally filling, root canal treatment, implants, and prostheses. Figure 1.1 shows how all these processes are viewed in a panoramic x-ray.

Figure 1.1. Panoramic Dental Image

The X-ray above shows us how an implant, filling, canal treatment and prosthesis looks like. An implant looks like a nail, a filling is like a single white piece, the canal treatment looks like a white line upwards or downwards, and the prosthesis looks like a combination of multiple fillings. These four treatments are objects on the x-ray that we need to detect and classify.

Deep learning is a machine learning method consisting of multiple layers that produce results with a given data set. Deep learning is a subset of machine learning that learns specific data on a particular subject. The main purpose of the deep learning algorithm is to do better each time. Deep learning can develop a solution in any area that requires a certain amount of thought and put it into practice. The relationship between artificial intelligence, machine learning, and deep learning is shown in Figure 1.2.

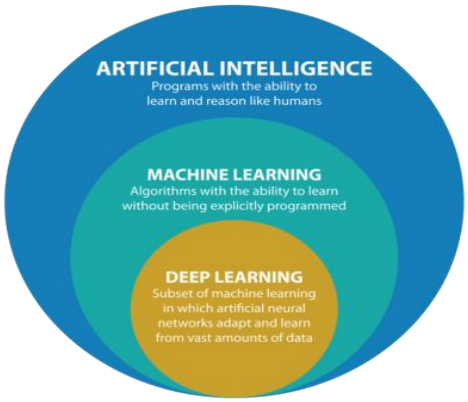

Figure 1.2. AI Venn Diagram(Raza & Cinquegrana, 2018)

Deep Learning, an artificial intelligence developed recently has increased the success of its applications to very high levels. Especially in the 2000s, it has been widely used thanks to the increasing amount of data and powerful GPU. High success rates have been achieved especially with Convolutional Neural Networks (CNN).

The use of the CNN algorithm starts with the extraction of a convolutional feature. Then, with the feature extracted from the image, classifier networks are used to recognize feature objects. Classifiers perform the object detection process by scanning the entire image in a sliding window depending on the selected algorithm.

There are two most important issues in object detection. The first is accuracy and the other one is speed. The accuracy and speed factors do not depend only on the chosen algorithm. It is seen that the larger the data set, the higher the rate of successful results in general. However, as a large data set means more images to be processed, it decreases the speed. Therefore, algorithm selection should be made by considering the speed and accuracy rate according to the available data set.

One of the oldest algorithms used in object detection is the R-CNN algorithm. R-CNN is one of the region-based object detection algorithms. First determines the areas where objects are likely to be found and then executes separate CNN classifiers there. Although this method gives good results, it reduces the speed as a picture is subjected to two separate processes.

Faster R-CNN algorithms have emerged as advanced versions to increase speed. These two algorithms became faster by passing the entire picture through the CNN structure at once and then sending it to the previously written Region Proposal Network, instead of the regions determined by segmentation separately. However, they are still too slow for speed-based studies. For this reason, YOLO is preferred in studies with priority speed. The reason why the YOLO algorithm is faster than R-CNN algorithms, in theory, is that it can predict the class and coordinates of all objects in the image by passing the image through the neural network in one go. Again, in theory, this speed provides an inverse proportion to the accuracy rate.

2. LITERATURE REVIEW

The use of deep learning methods, which are constantly evolving, has been quite high in recent years. Deep learning is frequently used in face recognition, voice recognition, security industry, defense industry and healthcare fields. And there are many studies in these areas in literature.

2.1. Convolutional Neural Networks

CNN is a type of multi-layer perceptron (MLP). The first CNN network in 1988 is the architecture named LeNet, proposed by LeCun (Lecun et al., 1998). In the LeNet network, the lower layers are successively placed convolution and maximum pooling consists of layers. The next upper layers are versus fully connected traditional MLP is coming. The architecture of LeNet is shown in Figure 2.1.

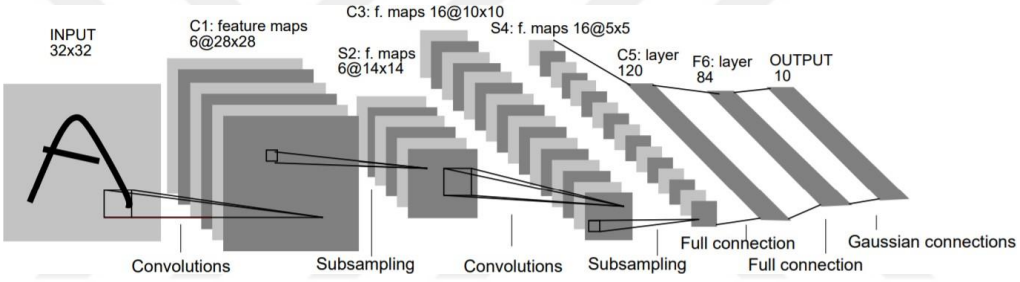

Figure 2.1. LeNet architecture (Lecun et al., 1998)

A very high degree of accuracy in computer vision was achieved with the CNN algorithm. In an article published in 2012 (Ciregan et al., 2012), the margin of error on the MNIST data set was reduced to 2%.In the same study, studies were made on other data sets (CIFAR10, traffic signs, NIST, SD 19) and recognition rates were increased on these data sets.

In 2014, ImageNet Contest, millions of pictures and hundreds of objects class and object classification and detection branch all of the teams with the most successful degrees are CNN have used modifications of their algorithm (Russakovsky et al., 2015). CNN reversals in a 2015 study faces in wide angle ranges including has shown the success of catching (Farfade et al., 2015). This network 200,000 images with faces in angles and directions, and 20 million more faceless images trained on the database.

CNN was also used for the Go game, and a pre-trained 12-layer CNN model beat the Go algorithm developed with traditional methods in 97% of the games (Maddison et al., 2015). Developed by Google DeepMind, the CNN-based AlphaGo is the first program that can beat a professional player (Lee et al., 2016).

2.2. Object Detection

In recent years, there are many studies in the field of object detection that have achieved high success rates using different methods. (Shen et al., 2019) worked on detecting vehicles from aerial images. In the study, the basis of the algorithm is the Faster R-CNN algorithm. The designed algorithm is applied to the Munich data set and the data set they have created. The system has achieved 90.2% accuracy rate.

Fast R-CNN algorithm is an advanced version of R-CNN algorithm (Girshick, 2015). Fast R-CNN employs several innovations to improve training and testing speed while also increasing detection accuracy. Fast R-CNN algorithm works 9 times faster than R-CNN algorithm. The biggest improvement in Fast R-CNN is the combination of CNN, SVM, and regressor used in R-CNN. With this combination, it achieves a tremendous performance advantage.

There are many several different works about object detection. (Li et al., 2016) aims to detecting and recognizing fish species from underwater images. In this work results show us Faster R-CNN methods give high accuracy they get 81.4% accuracies.

Single Shot Multibox Detector(SSD) was first published by (Liu et al., 2016). SSD algorithm emerged as a new option because R-CNN algorithms run very slow on large data sets and computers without powerful CPUs. The SSD algorithm performs all predictions on the CNN network at once, instead of performing different operations for each region to provide the required speed. However, although the SSD algorithm works fast, its accuracy rate is lower.

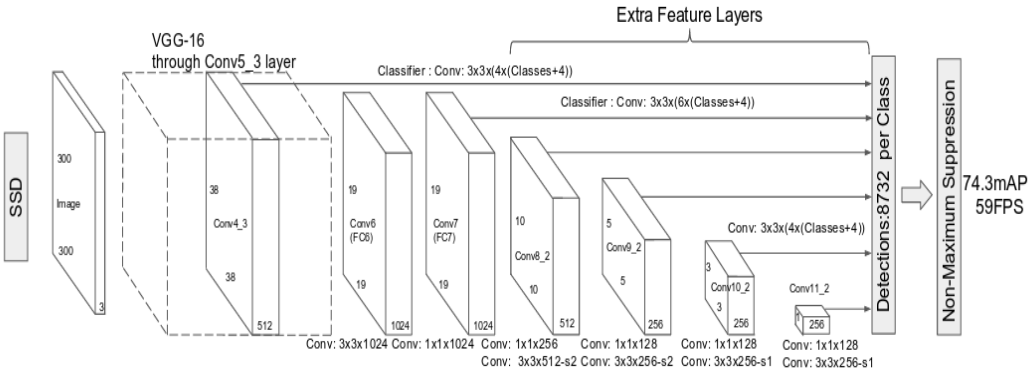

Figure 2.2. SSD architecture (Liu et al., 2016)

YOLO algorithm was first published by (Redmon et al., 2016). The most important feature that distinguishes YOLO algorithm from other object detection algorithms is that it can predict the class and coordinates of all objects in the picture by passing the picture through the neural network in one go. In other words, the basis of this estimation process deals with object detection as a single regression problem. This allows the YOLO algorithm to run faster than other algorithms. This makes the YOLO algorithm

indispensable for researchers who have real-time video data sets and do not have a powerful CPU.

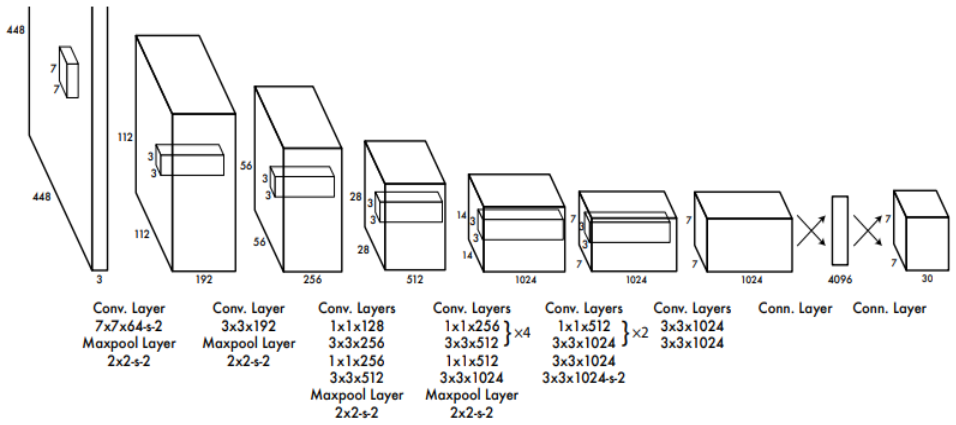

Figure 2.3. YOLO architecture (Redmon et al., 2016)

YOLOv3 is then presented as a version of the YOLO algorithm (Redmon & Farhadi, 2018). Although the old YOLO version ran very fast, it did not give confidence in terms of accuracy. YOLOv3 algorithm works three times faster than the SSD algorithm. It works slower than the old YOLO algorithm version. But the accuracy rate is higher.

2.3. Deep Learning in Healthcare

In the field of health, the priority for treatment is to diagnose the disease. One of the most preferred methods for diagnosis is x-rays. Many deep learning studies have been carried out in order to diagnose the disease from x-rays. (Wang et al., 2017) used the CNN algorithm to diagnose Pneumonia disease from X-rays. Later focused on RESNet50 to increase the accuracy rate. The highest accuracy rate in the study was 0.63 with the ResNet model.

In another study conducted on the same data set, the training was started with the weights of a predefined model in ImageNet, and the activation function sigmoid was used (Rajpurkar et al., 2017). They achieved 76% accuracy with their 121 layer CNN model named CheXNet.

In (Esteva et al., 2017), used CNN for the diagnosis of skin cancer. They showed that CNN trained with 129,450 clinical images made a classification with high accuracy. The authors achieved 72.1 ± 0.9% accuracy rate by using Google's Inception v3 architecture on the images they scaled 299x299.

In another study, they presented a DL-SVM-based hybrid model and CNN for melanoma diagnosis on dermoscopy images (Codella et al., 2015). In this study, they preferred the Caffe CNN model developed by Berkeley for CNN. In the experimental study conducted for melanoma and all lesions other than melanoma, a classification success of 91.9% was achieved with Caffe CNN and 93.1% with DL-SVM. In the study conducted for the differentiation of melanoma and atypical lesions, Caffe CNN achieved a classification success of 72.3% and DL-SVM 73.9%.

Deep learning methods have been used to detect the virus in the Covid-19 epidemic that has turned into a pandemic today. In this study, 98.08% accuracy rate for binary classifications and 87.02% accuracy rate for multi-class classifications using DarkNet framework and YOLO algorithm (Ozturk et al., 2020).

In another study about Covid-19, they reconstructed data classes on X-ray images using deep learning models. They achieved an accuracy rate of 99.27% for COVID-19 detection with optimization and chest X-ray images and stacking approaches using blurry colors (MesutToğaçar et al., 2020). They concluded that the model they proposed could be used effectively in detecting the virus in question.

In another study, COVID-19 disease was detected with CNN from the deep learning method by using 3905 X-Ray images belonging to seven classes (Apostolopoulos et al., 2020). The seven classes achieved an overall classification accuracy of 87.66% and 99.18% accuracy, 97.36% Sensitivity, and 99.42% specificity in the detection of COVID-19.

Another approach to detecting the COVID-19 virus from X-Ray is the Faster R-CNN framework. In this study proposed approach provides a classification accuracy of 97.36%, 97.65% of sensitivity, and a precision of 99.28% (HassanShibly et al., 2020).

2.4. Deep Learning in Dentistry

In (Lee et al., 2018) are working on their study which is evaluated the potential usefulness and accuracy of this system for the diagnosis and prediction of periodontally compromised teeth. With the CNN algorithm they reached 81% accuracies for premolars and 76.7% for molars.

In another the study the researcher experimented on the performance of CNN for diagnosis of a small labeled dental dataset and get 73% accuracies for CNN and 88% with transfer learning (Prajapati et al., 2017).

In (Takahashi et al., 2021) using the YOLO algorithm from 1904 oral photographic images, they tried to determine whether it was a prosthesis or a filling. As a result of this work but only 60% of tooth-colored prostheses were detected. The results of this study suggest that dental prostheses and restorations that are metallic in color can be recognized and predicted with high accuracy using deep learning; however, those with tooth color are recognized with moderate accuracy.

In another study (Motoki et al., 2020) tried to detect all teeth from a panoramic x-

ray using the Faster R-CNN. Working with a total of 1000 data sets, they have achieved 94% accuracy by using optimization methods.

3. MATERIAL AND METHOD

3.1. Data Set

3.1.1. Data Set Collection

The data set consists of panoramic x-ray images of patients with dental fillings, implants, prostheses, and canal treatment. All of the panoramic images and were taken from Akdeniz University Faculty of Dentistry. The data set consists of 313 panoramic x-rays and 1960 labels in total. Images are high resolution and in jpeg format. Every images are 2943 x 1435 and every images are from the same radiography machine. The x-ray images in the data set are labeled by dentists. Data imbalance, which is often encountered in studies in the field of health, has also been in this data set. The data set consists of more than 800 filler classes, more than 400 prosthesis classes, more than 200 canal treatment classes, and nearly 400 implant classes. Due to the widespread use of filling treatment in our country and the use of root canal treatment, it has also created an imbalance in our x-rays chosen randomly from the Akdeniz University Faculty of Dentistry. In Figure 3.1, fillings are shown in blue, prostheses in orange, root canal treatment in green, and the implant in red.

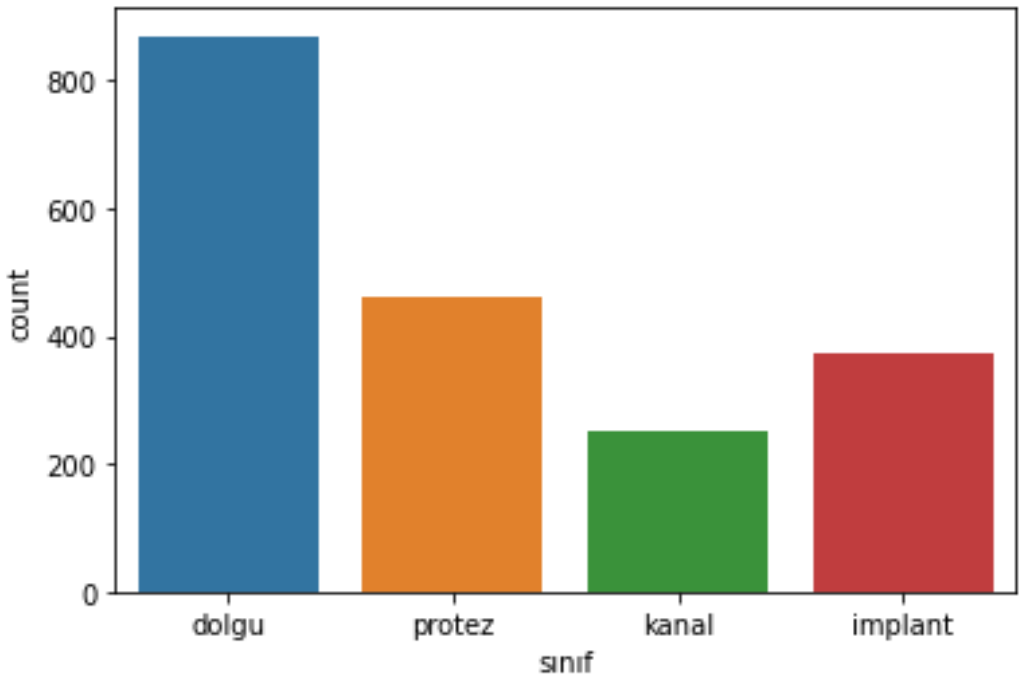

Figure 3.1. Distribution of Classes in Data Set

3.1.2. Labeling data set

Annotation technique means to label data that we have. Based on the algorithm used for detection there are different types of annotation techniques: bounding box

annotation, semantic annotation, contour annotation, and others. In this work, we will use bounding box annotation.

Before training YOLO, SSD, or Faster R-CNN, it is needed to prepare data set in their format. Every image has to have a text file with the same name as the image file has for YOLO for SSD or Faster R-CNN it has to be XML files. For instance, if we have an image with name image_1, then we need to have a text file with name image_1.txt. Inside this text file in one line for one object have the following: class number, object center in x, object center in y, object width and object height. Numbers for center point, object width, and object height need to be in a range from 0 to 1. That is why they are normalized by dividing on whole image width and whole image height respectively: object center x divided by image width, object center y divided by image height, object width divided by image width, object height divided by image height. In figure 3.2, how the text file with annotation for image looks like.

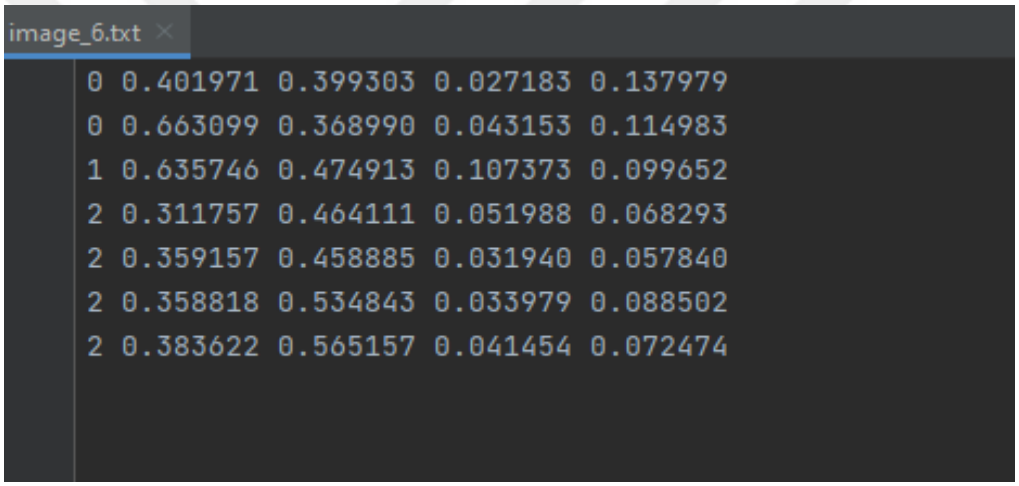

Figure 3.2. Txt file with annotation

There some useful resources for labeling images by bounding boxes. There a lot of online platforms and offline programs. In this work, LabelIMG program was used. LabelIMG is open source image labeling tool (Lin, 2018). In this program, results can save directly into YOLO format or XML files for SSD or Faster R-CNN algorithm. The LabelIMG program is very easy to install and very simple to use. It can be done to upload and annotate a single image or use the bulk uploading feature and open all images, going next and previous between them. It also allows an easy way to import and visualize already made annotations and correct them if necessary. It has a very simple and user-friendly offline interface that makes labeling process pretty fast. Figure 3.3 shows how to label in the LabelIMG program.

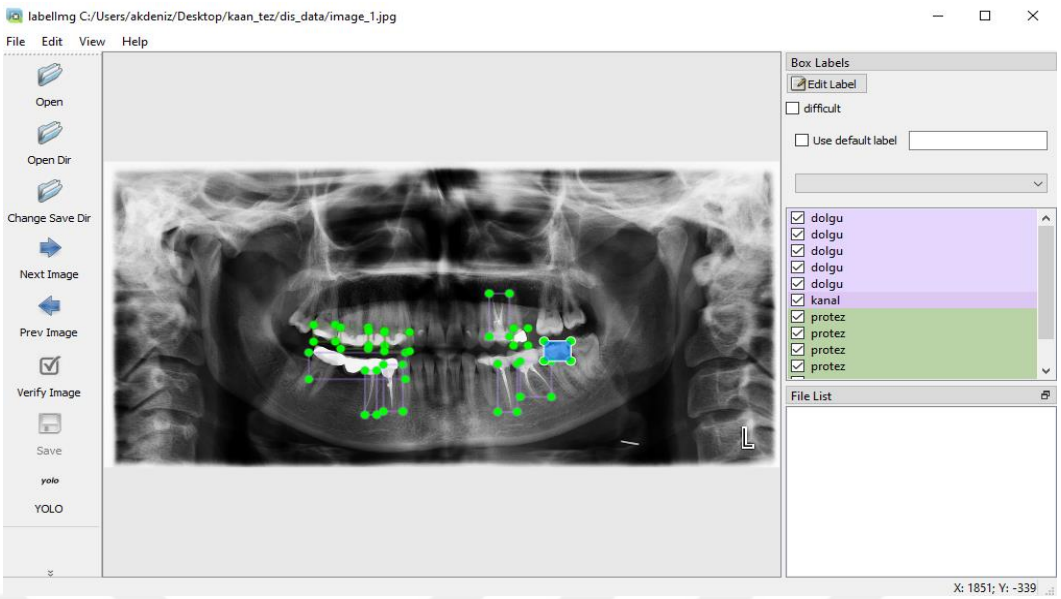

Figure 3.3. LabelIMG

3.2. Convolutional Neural Network

Convolutional Networks (ConvNets) are currently the most efficient deep models for classifying image data. Their multistage architectures are inspired by the science of biology. Through these models, invariant features are learned hierarchically and automatically (Jmour et al., 2018). If you want to classify or recognize a picture with traditional neural networks, all pixels must be transferred to the neural network. In convolutional networks, first of all, some patterns are tried to be detected on the picture, and these patterns are transferred to the neural network. In this way, while we can process the picture in a less complex way, we can achieve more successful results. The pictures given as input must be recognized by computers and converted into a format that can be processed. For this reason, pictures are first converted to matrix format. The system determines which picture belongs to which label based on the differences in pictures and therefore in matrices. He learns the effects of these differences on the label during the training phase and then makes predictions for new pictures using them. CNN has some layers to perform these operations effectively. These layers will be examined sequentially.

It is complete CNN model steps: “conv => max pool => dropout => conv => max pool => dropout => fully connected” for 2 layers CNN.

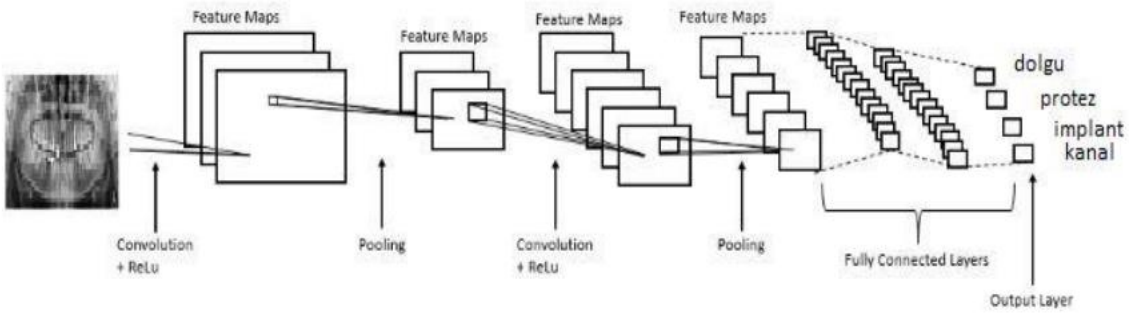

Figure 3.4. Complete CNN model

3.2.1. Feature extraction

Without careful attention to all the details in a picture we cannot understand what the picture is about. It is essential for the computer to be able to recognize that picture so that it can detect all the details in a photo. Because of that we use CNN. CNN extracts features on convolution layer. We can say that convolution operation. Center element of the kernel placed over the source pixel. The source pixel is then replaced with a weighted sum of itself and nearby pixels.

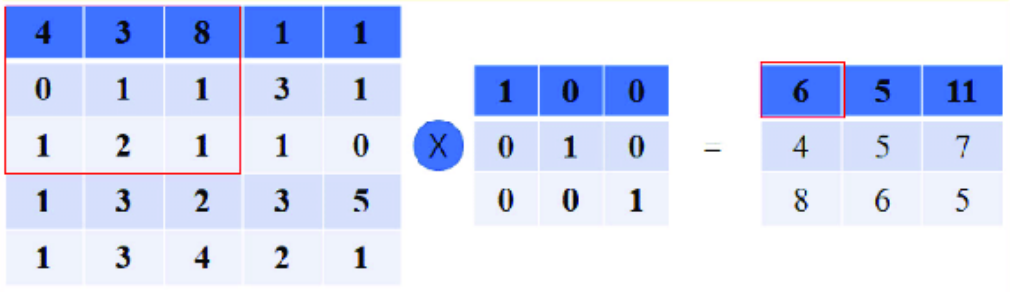

Figure 3.5. Convolution Operation (Wang et al., 2019)

3.2.2. Padding

After convolution operation the new size of the image smaller than the original image size. As a result of this operation, there is data loss. The process to prevent this data loss is called the padding process. There are different padding methods. Same padding, valid padding, constant padding, causal padding, reflection padding, and replication padding some of them. In case the input matrix nxn is the filter (weight) matrix (fxf), if the output matrix is desired to be the same size as the input; the formula (n + 2p-f + 1) x (n + 2p-f + 1) is applied. The value shown with 'p' here is the pixel size added to the input matrix, namely the padding value. To determine this, p = (f-1) / 2 equation is used.

3.2.3 Pooling and stride

The size protection process in the padding part is important to prevent data loss. However, large sizes also cause the system to run slowly. First, let's create a 2 × 2 filter. You can see this filter on the (4 × 4) Figure 3.6 below. As you can see in the picture, the filter takes the largest number in the area it covers. In this way, it uses smaller outputs that contain enough information for the neural network to make the right decision. The name of this process is max pooling. There are average pooling and L2-norm pooling algorithms that work on the same principle. Also, pooling prevents overfitting. However, many people do not prefer to use this layer. Instead, the larger Stride is preferred in the Convolutional layer. The stride process informs that it will shift the filter, which is the weight matrix, on the image in steps of one pixel or larger steps for the convolution process. This is another parameter that directly affects the output size. For example, when the padding value is p = 1, the size of the output matrix when the number of steps s = 2 is selected if the formula (((n+2p-f)/s)+1)x(((n+2p-f)/s)+1) is applied, if it is calculated for the values n = 5 and f = 3, the output size becomes (3) x (3).

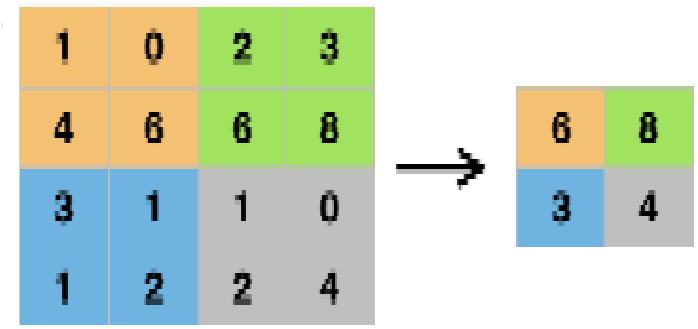

Figure 3.6. Pooling operation

3.2.4. Fully-Connected Layer

This layer is the last and most important layer of CNN architecture. Figure 3.7 shows a simple neural network structure. This structure can also be seen as a simpler version of the fully connected layer structure. Consequently, understanding this structure will help us understand the working principle of the fully connected layer. First of all, the input layer part that we get by flattening with x1 and x2 is created. Then the values from the input layer are processed with h1 and h2 and hidden layers, which we call hidden layers, with the coefficients determined by the model (often the function). According to the activation function determined as a result of this process, y, ie output value is produced. The fully connected layers learn a (possibly non-linear) function between the high-level features given as an output from the convolutional layers.

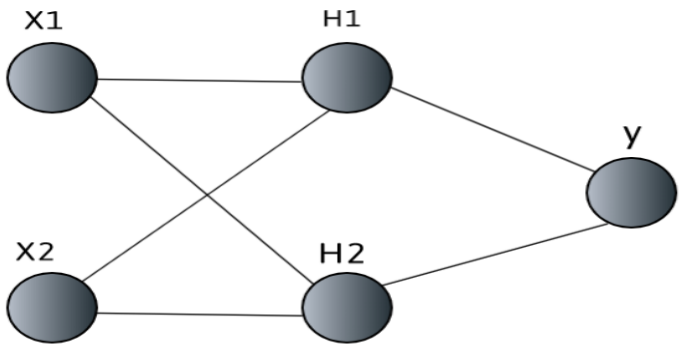

Figure 3.7. Fully-Connected Layer

3.3. Faster R-CNN Model

In Faster R-CNN, the detection process is carried out in two stages (Fuentes et al., 2017). In the first stage, a Region Proposal Network (RPN) takes an image as input and processes it by a feature extractor. Features at an intermediate level are used to predict object proposals, each with a score. For training the RPNs, the system considers anchors containing an object or not, based on the Intersection-over-Union (IoU) between the object proposals and the ground-truth. In the second stage, the box proposals previously generated are used to crop features from the same feature map. Those cropped features are consequently fed into the remaining layers of the feature extractor in order to predict the class probability and bounding box for each region proposal.

The entire process happens on a single unified network, which allows the system to share full-image convolutional features with the detection network, thus enabling nearly cost-free region proposals. Since the Faster R-CNN was proposed, it has influenced several applications due to its outstanding performance on complex object recognition and classification.

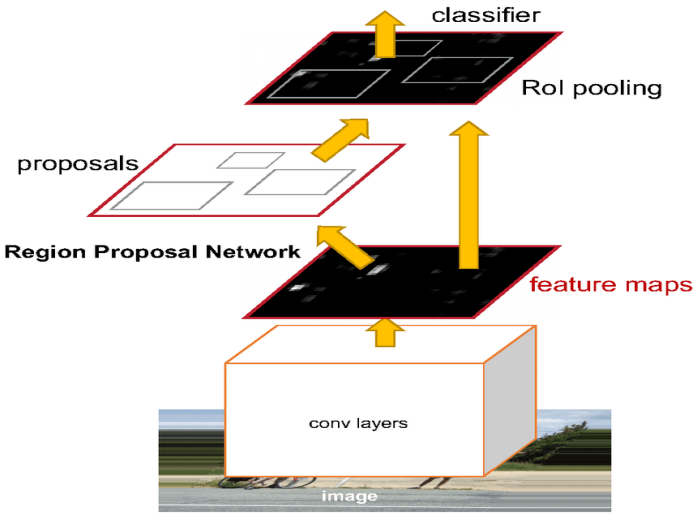

Figure 3.8. Faster R-CNN

3.3.1. Tensorflow object detection API

The TensorFlow Object Detection API is an open-source framework built on top of TensorFlow that makes it easy to construct, train and deploy object detection models.

The Tensorflow Object Detection API offers us some models which trained on the coco dataset. In the coco dataset, there are 90 different classes. For training, one of these models should be selected. When choosing, care should be taken to choose a model that is suitable for its own data set. In this study, a model named faster_rcnn_inception_v2_coco_2018_01_28 is used in terms of compatibility with our data set. The names of other available models, their average speeds, and their success rates are shown in figure 3.8.

Model name	Speed (ms)	COCO mAP[^1]	Outputs
ssd_mobilenet_v1_coco	30	21	Boxes
ssd_mobilenet_v1_0.75_depth_coco ☆	26	18	Boxes
ssd_mobilenet_v1_quantized_coco ☆	29	18	Boxes
ssd_mobilenet_v1_0.75_depth_quantized_coco ☆	29	16	Boxes
ssd_mobilenet_v1_ppn_coco ☆	26	20	Boxes
ssd_mobilenet_v1_fpn_coco ☆	56	32	Boxes
ssd_resnet_50_fpn_coco ☆	76	35	Boxes
ssd_mobilenet_v2_coco	31	22	Boxes
ssd_mobilenet_v2_quantized_coco	29	22	Boxes
ssdlite_mobilenet_v2_coco	27	22	Boxes
ssd_inception_v2_coco	42	24	Boxes
faster_rcnn_inception_v2_coco	58	28	Boxes
faster_rcnn_resnet50_coco	89	30	Boxes
faster_rcnn_resnet50_lowproposals_coco	64		Boxes
rfcn_resnet101_coco	92	30	Boxes
faster_rcnn_resnet101_coco	106	32	Boxes

Figure 3.9. Coco Trained Models

3.3.2. Training Procedure

Tensorflow application program interface (API) is the technique that employs CNN models to draw boxes around the region of interest (ROI). The process of applying API is given in the following steps.

1) We mentioned that we created XML files in the data set section.Now these XML files along with the associated image are given to the xml_to_csv python script. This script merges all the XML files into one CSV file.

2) The CSV file obtained in step 1 is then given to the tfrecord maker python script.

3) The CSV file, along with the associated tfrecord file is then given as inputs to the training algorithm which trains the model to recognize the pattern and other features of the teeth.

4) The trained model is then used for the testing purpose i.e., to draw the boxes around the teeth is present in the test images.

3.4. YOLO Model

YOLO uses CNN for object detection. YOLO can detect multiple objects on a single image. It means that apart from predicting classes of the objects, YOLO also detects the locations of these objects on the image. YOLO applies a single neural network to the whole image. This neural network divides the image into regions and produces probabilities for every region. After that, YOLO predicts the number of bounding boxes that cover some region on the image and chooses the best ones according to the probabilities. In this study, YOLO's v3 and v4 updates were tested separately on the data set for object recognition.

3.4.1. YOLOv3

YOLOv3, originally, consists of 53 convolutional layers that are also called Darknet-53. But for the detection task, the original architecture is stacked with 53 more give us 106 layers of architecture for YOLOv3. The process of loading that consists of 106 layers in the darknet framework is shown in Figure 3.8.

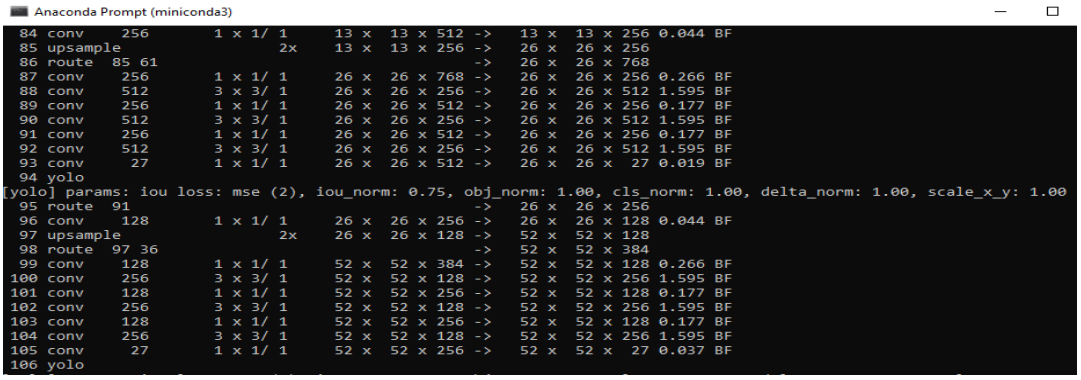

Figure 3.10. YOLOv3 106 layers

YOLOv3 consists of 103 convolutional layer and 3 yolo layers. Each convolutional layers is followed by the batch normalization layer and Leaky ReLU activation function. As mentioned in the section where we overview the convolutional layer network section, in YOLO there are no pooling layers, but instead, additional convolutional layers with stride 2 are used to down-sample feature maps. In this way, it is very successful in detecting small objects on the image. Thanks to this feature, YOLOv3 is very good at detecting very small objects in our data set.

YOLOv4 has been released in April 2020(Bochkovskiy et al., 2020). YOLOv4 unlike YOLOv3 has the implementation of new architecture in the Backbone and the modifications in the Neck. Thanks to these modifications and new architecture, YOLOv4 has improved the mAP (mean Average Precision) by 10% and the number of FPS (Frame per Second) by 12%.

YOLOv3 and YOLOv4 also have other versions with smaller layers called YOLOv3-tiny and YOLOv4-tiny. YOLOv3-tiny's backbone network only includes 7 convolutional layers and 6 max-pooling layers, and its feature pyramid network is also simplified by removing the maximum-scale prediction branch and reducing the number of convolutional layers in the other two branches. The YOLOv4-tiny model consists of 38 layers in total. Both algorithms are usable algorithms because they will work very quickly for deep learning projects to be classified less.

3.4.2. Setting up Parameters of YOLO

YOLOv3 makes detections at three different scales and three spate places in the network. These separate places for detections are layers 80, 94, and 106 named yolo. Network down samples, input image by following factors: 32, 16, and 8 at those separate places of accordingly. These three numbers are called stride to the network, and they show how the output at three separate places in the network is smaller than the input to the network. Since the purpose of this study is to detect classes of small sizes from x-rays input network size 416 by 416, stride 8 has been selected. In this way, the size of the

outputs has been 52 by 52.

To produce output, YOLOv3 applies a 1 by 1 detection kernels at these three separate places in the network. 1 by 1 convolution applied to down sampled input images: 13 by 13, 26 by 26 and 52 by 52. Consequently, resulted from feature maps will have the same spatial dimensions. The shape of the detection kernel also has its depth that is calculated by the following equation; (b x (5+ c)) b here represents the number of bounding boxes that each cell of the produced feature map can predict. YOLOv3 predicts three bounding boxes for every cell of these future maps. That is why b is equal to three. In the data set, we have 4 different classes and c in the formula represents the number of classes of data set. The resulted equation is as following: 3 multiplied by 9, which gives us 27 attributes. Now we calculated our feature maps shape as following (52, 52, 27).

To predict bounding boxes, YOLOv3 uses pre-defined default bounding boxes that are called anchors or priors. These anchors are used later to calculate the predicted bounding box's real width and real height. In total 9 anchor boxes are used. Three anchor boxes for each scale 80, 94, and 106. To calculate these anchors, K-means clustering is applied in YOLOv3.

Next, it is needed to change max_batches that is a total number of iterations for training and steps that are used for updating the learning rate. max_batches is updated according to the number of classes. General equation is as following: max_batches = classes * 2000 in this study max_batches = 8000. Steps are calculated as 80% and 90% from max_batches. In this study steps = 6400,7200.

4. RESULTS AND DISCUSSIONS

4.1. Experiment Details

Many mathematical operations need to be calculated in the train and test phases of a deep learning application. Factors such as the size of the dataset and the number of layers of the model cause this process to take hours and days. This is why a powerful GPU is essential for a deep learning application. In this study, Google's cloud artificial intelligence development system which is called "Google Colab" was used. In this study where many different models were tried, time was saved thanks to the powerful CPU provided by Colab. Colab has 2 versions. One of them is free and the other is the Pro version. The Pro version was chosen as the free version offers time restrictions and a weaker CPU than the Pro version. Table 4.1 shows the System Features provided by Google Colab Pro.

Table 4.1. System Features provided by Google Colab Pro(Anonymous)

GPU Name	Nvidia Tesla V100 GPU
GPU Memory	16 GB RAM

Factors affecting the success of the deep learning model are the number of iterations, correct labelling, and the size of the data set, as well as the model used. There are some methods used to grow a data set. All these methods are called data augmentation. Data augmentation techniques in the field of computer vision are mainly: adding noise, cropping, rotation, scaling, translation, brightness, colour augmentation and saturation. Some of the images in the train data were randomly selected and reproduced by horizontal flip process, crop operation at minimum 1% and maximum 20%, rotation between -15 ° and + 15 ° and adding some noise. These operations are also applied to some images together. Thanks to the applied data augmentation techniques, the number of images in the train folder has increased to two times. Some images with data augmentation are shown in Figure 4.1.

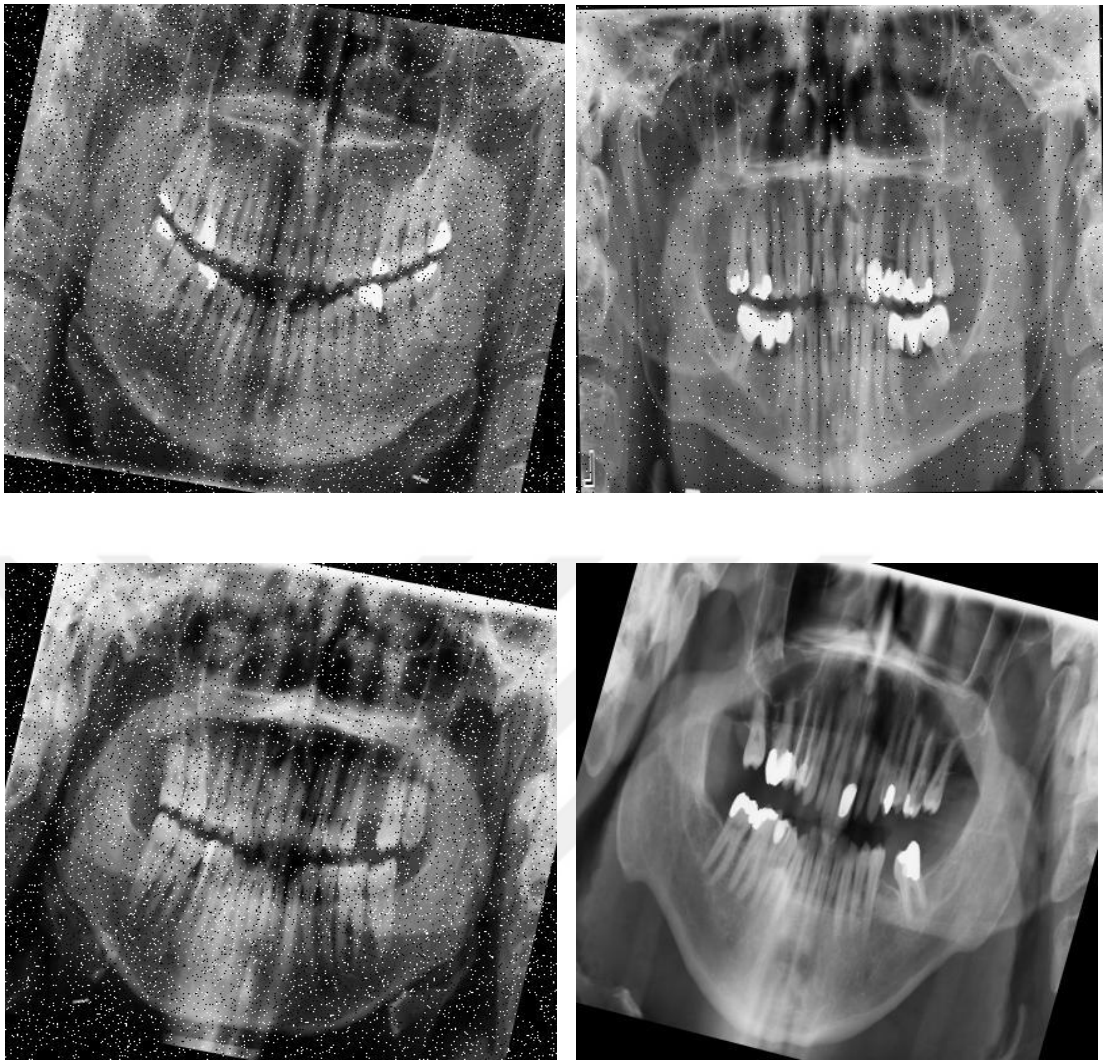

Figure 4.1. Images with data augmentation

In this study, the Darknet framework is used for YOLO models(Anonymous). Also, Tensorflow and Keras open-source artificial intelligence libraries were used for creating and developing for Faster-R-CNN model (Anonymous).

4.2. Experiment Results

4.2.1. Precision, Recall, F1 score and Map

It is wrong to decide whether a model created for classification in the field of deep learning is successful or not only based on accuracy rate. The Confusion Matrix table shown in Figure 4.2 shows the realized and predicted values in a classification problem.

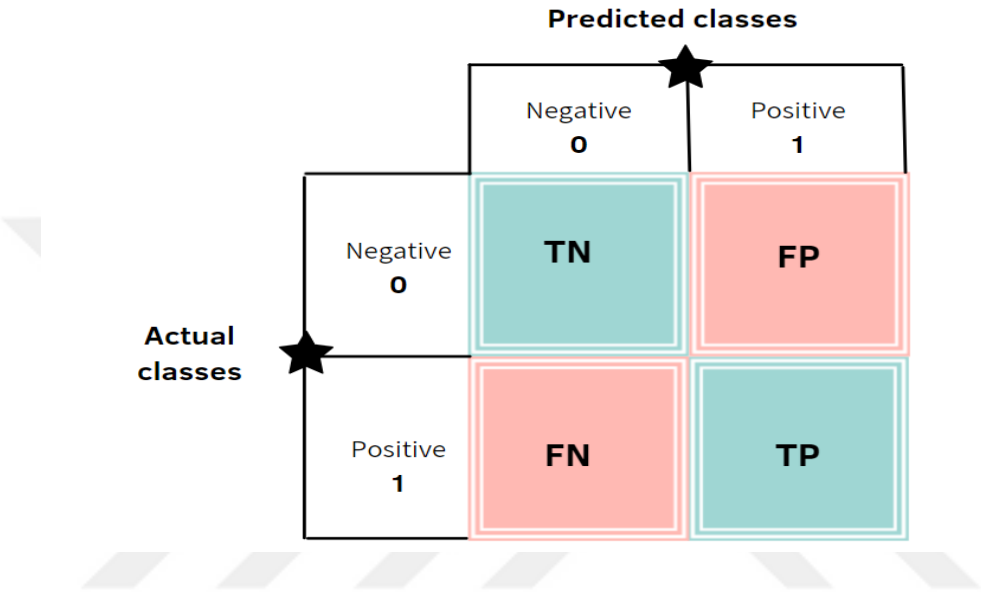

Figure 4.2. Confusion matrix

True Positive (TP) and True Negative (TN) are areas that the model predicts correctly, while False Positive (FP) and False Negative (FN) are areas that the model predicts incorrectly. In this study, if our model predicts an implant as an implant and this is true, then this is TP. If the model guesses that this is not an implant, and this is true, then this is TN. The model predicts as an implant, and if this is incorrect, it is referred to as type 1 error and this is an FP. The model predicts not an implant and if it is wrong it is also called a type 2 error and it is an FN. The Confusion matrix is used to calculate precession, recall, F1 score and mAP values.

Precision is calculated by the ratio of correct positive results in predicted samples to all results predicted positively. The formula for the precision value is shown in Formula 4.1.

$$P = \frac{TP}{TP+FP} \qquad (4.1)$$

The recall is calculated as the ratio of correct positive results in predicted samples to all positive results that should be. The formula for the sensitivity value is shown in Formula 4.2.

$$R = \frac{TP}{TP+FN} \quad (4.2)$$

F1 score is calculated as the harmonic mean of the values given above. The formula for the F1 score is shown in Formula 4.3.

$$F1\ Score\ = 2 \times \frac{P \times R}{P + R} \quad (4.3)$$

mAP is a metric system designed to evaluate values such as precision, sensitivity and F1 score from a single point. The formula for the mAP value is shown in Formula 4.4.

$$mAP = \int_0^1 P(R)dR \quad (4.4)$$

4.2.2. Loss function

The model checks for errors for its predictions and tries to minimize the error continuously. For this, it tries to calculate the error and converge to the expected result, in other words, to reduce the error. While calculating the error with the Loss function, the optimizer tries to find the least probability of error. For a model to increase its accuracy, the loss function value must be close to 0. Another importance of the loss function shows us when we should end the train phase. A model starts with a high loss function value and improves itself over time and decreases the loss function value. However, after a certain number of iterations, the model cannot develop itself further and the loss function value starts to go stable. In this way, we understand that the train phase should stop and the model can no longer be trained. Again, thanks to the loss function, we understand that if we predetermined how many iterations our model will make, but when the train phase is completed, if the loss function value is still in a downward trend, we need to increase our iteration number.

4.3. Results of models

4.3.1. YOLOv3-tiny model

The training with the YOLOv3 - tiny model took a total of 2 hours and 26 minutes. Loss function value decreased to 3.51 as shown in Figure 4.3.

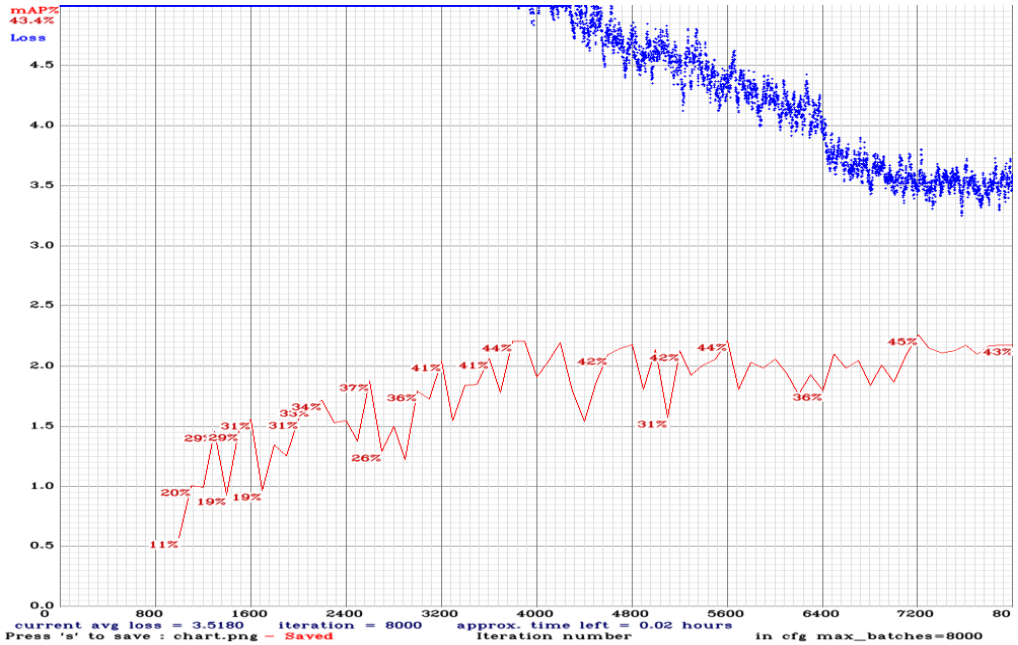

Figure 4.3. YOLOv3 – tiny loss function

Precision value was calculated as 0.49, Recall value as 0.53, F1 score value as 0.51 mAP value was as 0.43. Total detection time was calculated as less than 1 second. The values calculated as a result of the training are shown in Figure 4.4.

```
 detections_count = 1312, unique_truth_count = 395
class_id = 0, name = dolgu, ap = 45.08%          (TP = 112, FP = 113)
class_id = 1, name = implant, ap = 37.23%        (TP = 53, FP = 51)
class_id = 2, name = kanal, ap = 23.78%          (TP = 22, FP = 42)
class_id = 3, name = protez, ap = 67.55%         (TP = 21, FP = 9)

 for conf_thresh = 0.25, precision = 0.49, recall = 0.53, F1-score = 0.51
 for conf_thresh = 0.25, TP = 208, FP = 215, FN = 187, average IoU = 31.68 %

 IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
 mean average precision (mAP@0.50) = 0.434121, or 43.41 %
Total Detection Time: 0 Seconds
```

Figure 4.4. YOLOv3 – tiny Precision, Recall, F1 score and mAP values

As a result, it has been quite unsuccessful in identifying classes. The result is shown in Figure 4.5.

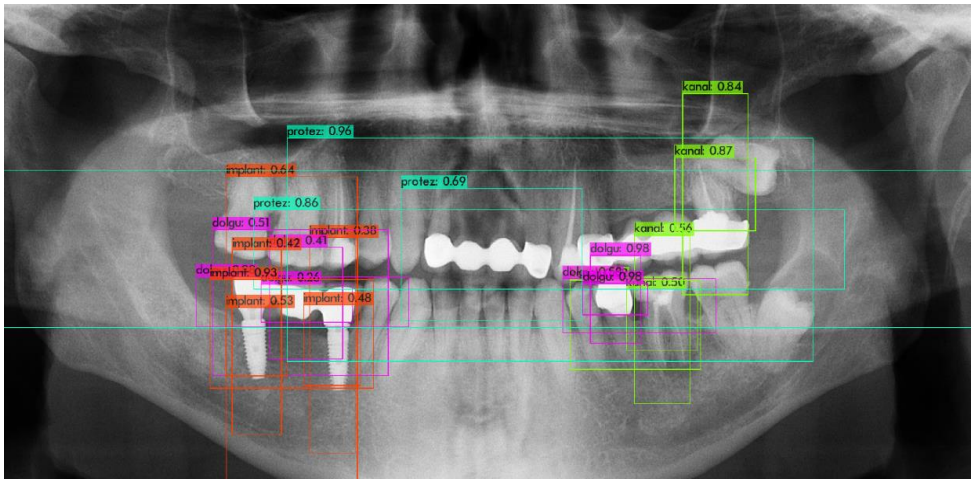

Figure 4.5. YOLOv3 – tiny result

4.3.2. YOLOv3 model

The training with the YOLOv3 model took a total of 6 hours and 48 minutes. Loss function value decreased to 0.66 as shown in Figure 4.6.

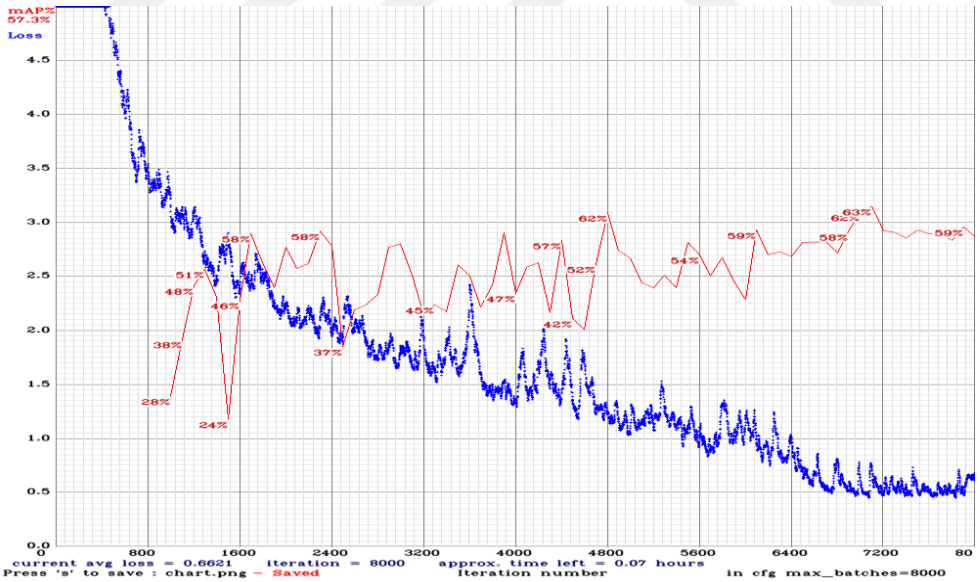

Figure 4.6. YOLOv3 loss function

Precision value was calculated as 0.68, Recall value as 0.60, F1 score value as 0.63 mAP value was as 0.57. Total detection time was calculated as less than 1 second.

The values calculated as a result of the training are shown in Figure 4.7.

```
60
 detections_count = 538, unique_truth_count = 395
class_id = 0, name = dolgu, ap = 48.31%          (TP = 114, FP = 65)
class_id = 1, name = implant, ap = 60.87%        (TP = 69, FP = 24)
class_id = 2, name = kanal, ap = 42.32%          (TP = 31, FP = 14)
class_id = 3, name = protez, ap = 77.76%         (TP = 22, FP = 10)

 for conf_thresh = 0.25, precision = 0.68, recall = 0.60, F1-score = 0.63
 for conf_thresh = 0.25, TP = 236, FP = 113, FN = 159, average IoU = 45.37 %

 IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
 mean average precision (mAP@0.50) = 0.573189, or 57.32 %
Total Detection Time: 0 Seconds
```

Figure 4.7. YOLOv3 Precision, Recall, F1 score and mAP values

As a result, it has been observed that our model is successful in determining the classes but does not give a high accuracy rate to the determinations made. The result is shown in Figure 4.8.

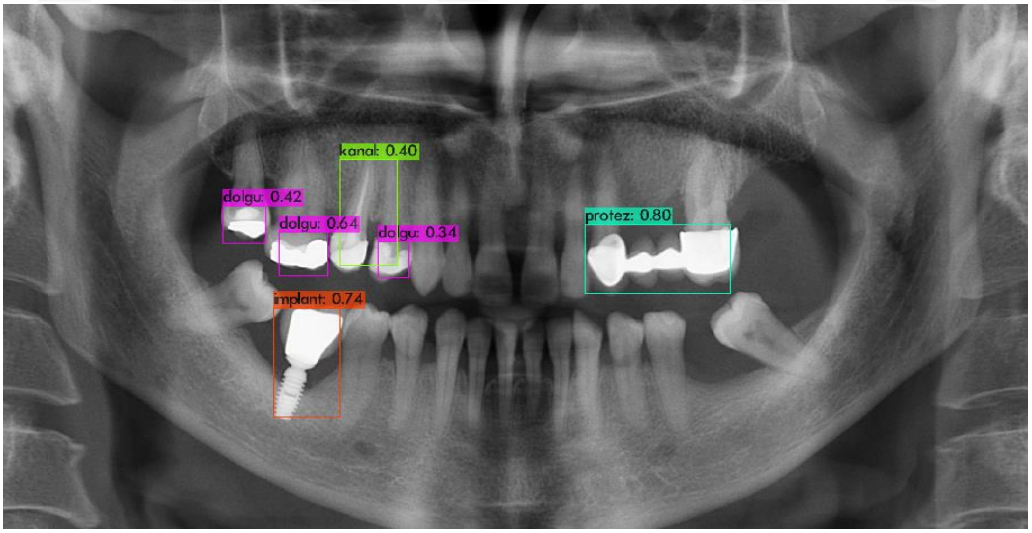

Figure 4.8. YOLOv3 result

4.3.3. YOLOv4-tiny model

The training with the YOLOv4 - tiny model took a total of 3 hours and 10 minutes.

Loss function value decreased to 2.33 as shown in Figure 4.9.

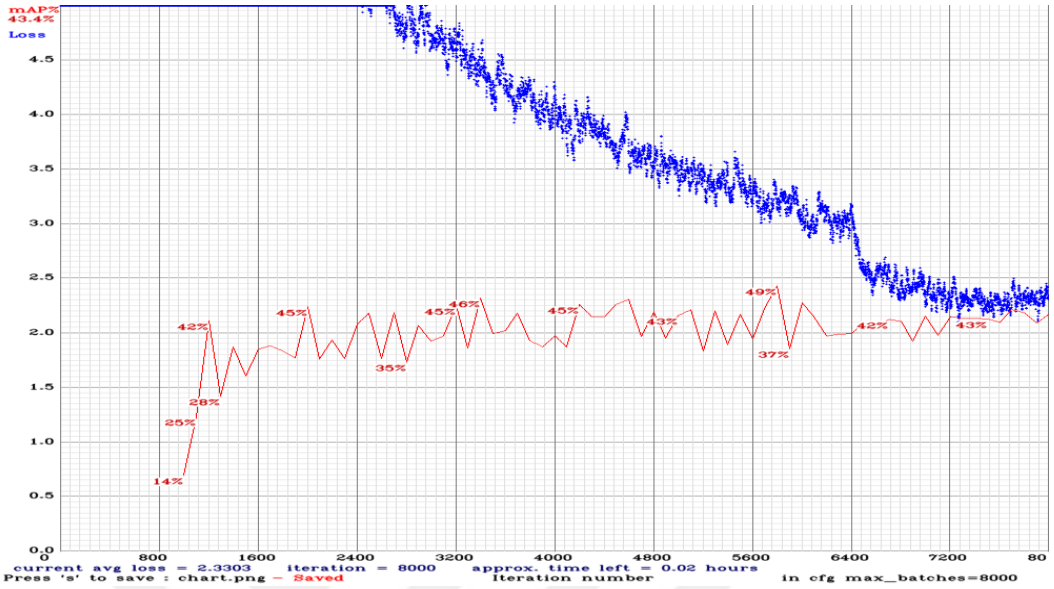

Figure 4.9. YOLOv4 – tiny loss function

Precision value was calculated as 0.50, Recall value as 0.55, F1 score value as 0.53 mAP value was as 0.43. Total detection time was calculated as high than 1 second. The values calculated as a result of the training are shown in Figure 4.10.

```
 detections_count = 947, unique_truth_count = 395
class_id = 0, name = dolgu, ap = 44.22%          (TP = 112, FP = 109)
class_id = 1, name = implant, ap = 33.28%        (TP = 52, FP = 54)
class_id = 2, name = kanal, ap = 37.79%          (TP = 34, FP = 40)
class_id = 3, name = protez, ap = 58.39%         (TP = 21, FP = 12)

 for conf_thresh = 0.25, precision = 0.50, recall = 0.55, F1-score = 0.53
 for conf_thresh = 0.25, TP = 219, FP = 215, FN = 176, average IoU = 32.90 %

 IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
 mean average precision (mAP@0.50) = 0.434231, or 43.42 %
Total Detection Time: 1 Seconds
```

Figure 4.10. YOLOv4 – tiny Precision, Recall, F1 score and mAP values

As a result, it has been observed that the model failed to detect the classes and made many incorrect determinations. The result is shown in Figure 4.11.

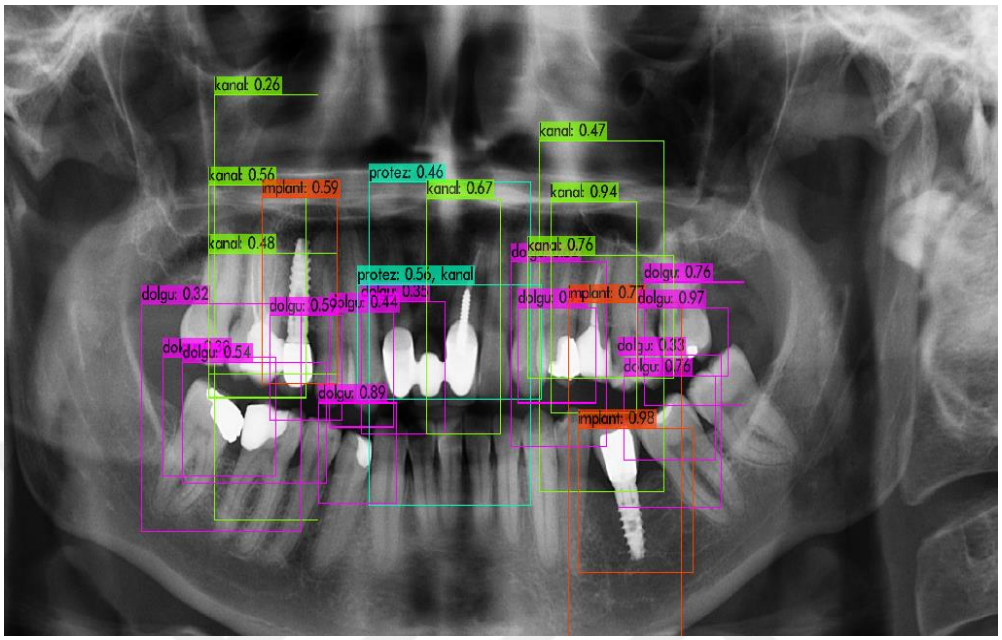

Figure 4.11. YOLOv4 – tiny result

4.3.4. YOLOv4 model

The training with the YOLOv4 model took a total of 9 hours and 22 minutes. Loss function value decreased to 1.03 as shown in Figure 4.12.

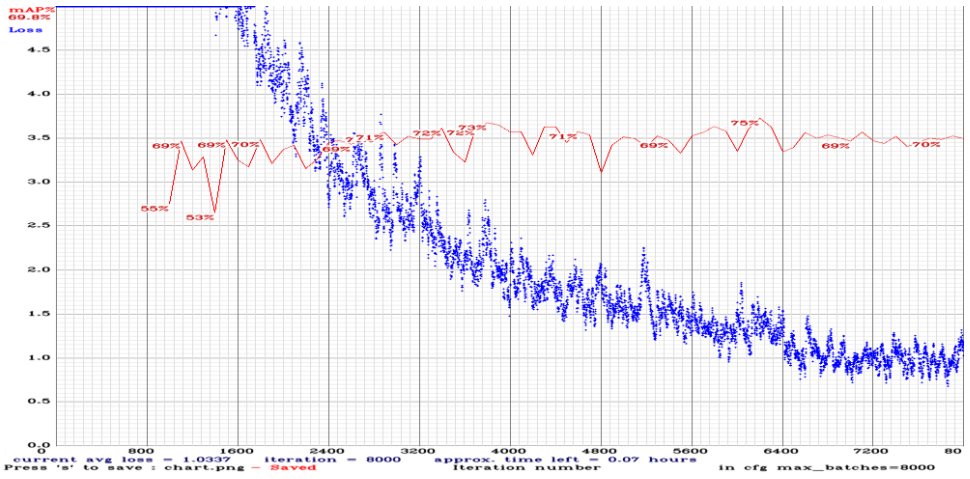

Figure 4.12. YOLOv4 loss function

Precision value was calculated as 0.76, Recall value as 0.68, F1 score value as 0.72 mAP value was as 0.70. Total detection time was calculated as high than 1 second. The values calculated as a result of the training are shown in Figure 4.13.

```
 calculation mAP (mean average precision)...
60
 detections_count = 471, unique_truth_count = 395
class_id = 0, name = dolgu, ap = 53.69%          (TP = 117, FP = 50)
class_id = 1, name = implant, ap = 75.15%        (TP = 79, FP = 14)
class_id = 2, name = kanal, ap = 75.18%          (TP = 48, FP = 12)
class_id = 3, name = protez, ap = 75.15%         (TP = 23, FP = 7)

 for conf_thresh = 0.25, precision = 0.76, recall = 0.68, F1-score = 0.72
 for conf_thresh = 0.25, TP = 267, FP = 83, FN = 128, average IoU = 51.93 %

 IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
 mean average precision (mAP@0.50) = 0.697945, or 69.79 %
Total Detection Time: 1 Seconds
```

Figure 4.13. YOLOv4 Precision, Recall, F1 score and mAP values

As a result, it has been observed that the model is quite successful in determining the classes and it gives a very high success rate to the determinations made. The result is shown in Figure 4.14.

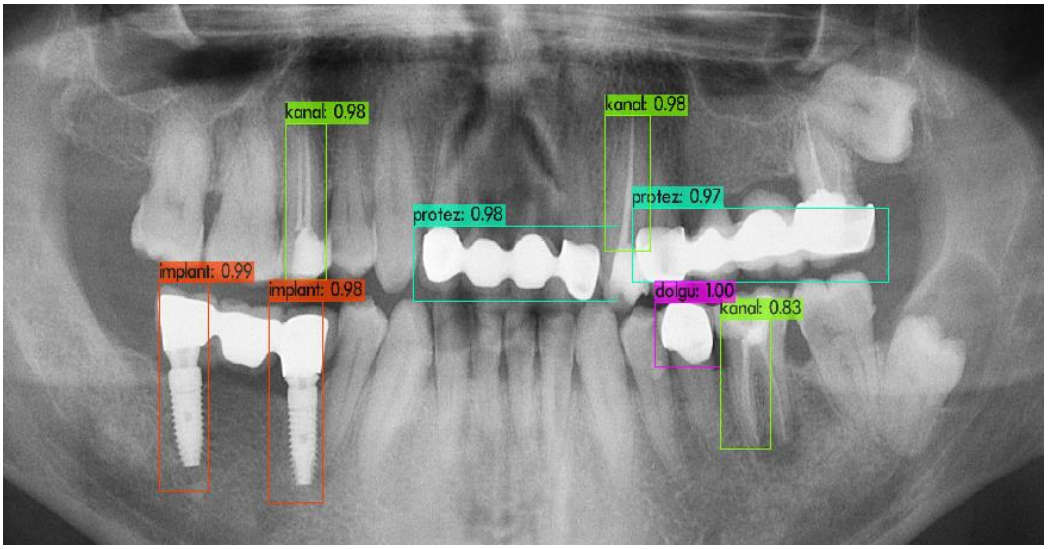

Figure 4.14. YOLOv4 result

4.3.5. Faster R-CNN model

The training with the YOLOv4 model took a total of 11 hours and 34 minutes. Loss function value decreased to 0.4 as shown in Figure 4.15.

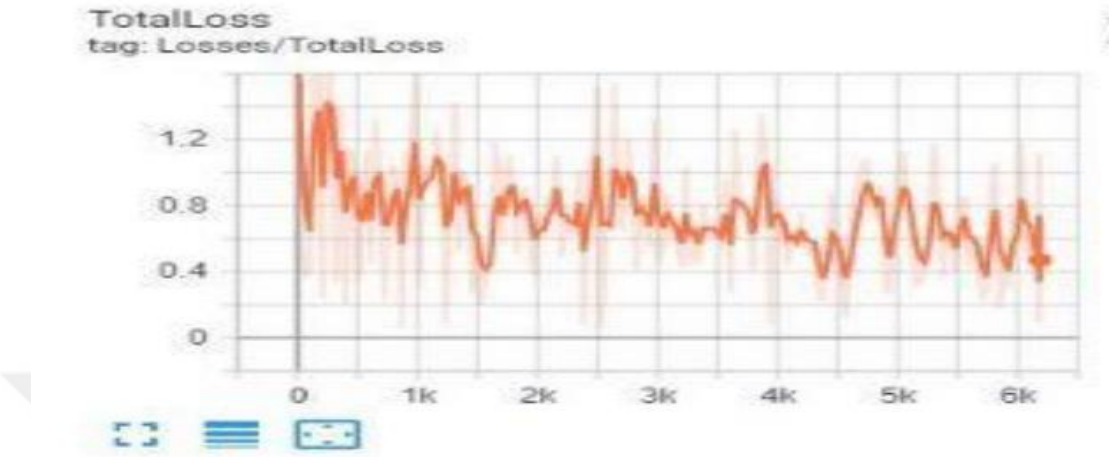

Figure 4.15. Faster R-CNN loss function

Precision value was calculated as 0.71, Recall value as 0.64, F1 score value as 0,67 mAP value was as 0.63. Total detection time was calculated as high than 3 seconds. As a result, it has been observed that the model is quite successful in determining the classes and it gives a very high success rate, however, it has also been observed that could not detect some classes. The result is shown in Figure 4.16.

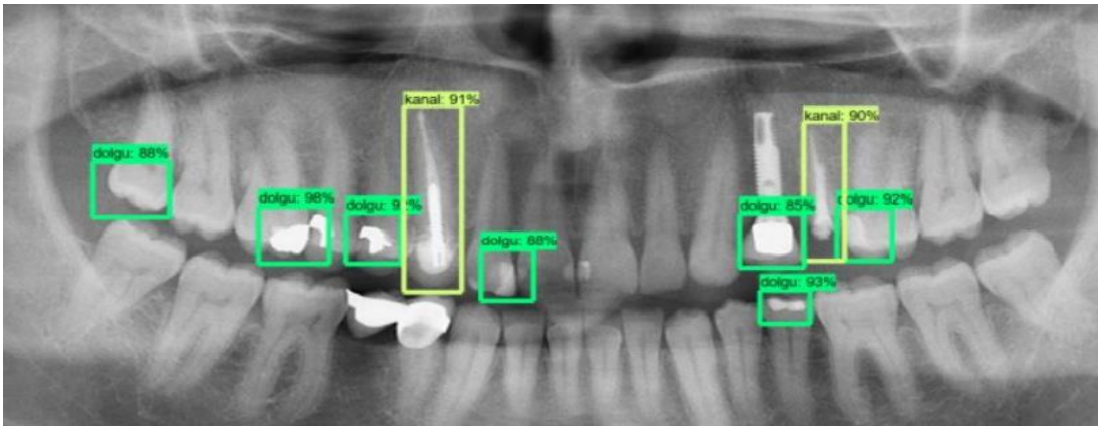

Figure 4.16. Faster R-CNN result

4.4. Comparison of models

All the train and test stages in this study were carried out on the same data set and by the same GPU. Although YOLO - tiny models with fewer layers are successful in speed, it has been determined that they cannot make a meaningful classification. Successful classification results have been provided in YOLOv3, YOLOv4, and Faster R-CNN models. Among these models, the fastest results are taken from the YOLOv3 model. The most successful results are from the YOLOv4 model. The comparison of all the results obtained is made in Table 4.2.

Table 4.2. Comparison of models

Model name	YOLOv3-tiny	YOLOv3	YOLOv4-tiny	YOLOv4	Faster-R-CNN
Precesion	0.49	0.68	0.50	0.76	0.71
Recall	0.53	0.60	0.55	0.68	0.64
mAP	0.43	0.57	0.43	0.70	0.63
F1 score	0.51	0.63	0.53	0.72	0.67
Loss function	3.51	0.66	2.33	1.03	0.40
FPS	200	65	170	25	7
Speed	8ms/img	14ms/img	9ms/img	32ms/img	140ms/img

5. CONCLUSION

The aim of this study is to develop a fast and highly accurate model that can assist dentists in diagnosing. In this context, X-rays were collected and labelled to create our data set, taking care to include filling, prosthesis, implant, and root canal treatment processes. Later, data augmentation techniques were applied to this data set and the data set was reproduced and made ready to use to train our model.

The most suitable algorithms for the data set were searched by analysing the literature studies. It has been observed in the studies that YOLO versions and Faster R-CNN algorithm are widely used and achieved successful results. Considering that the study is in the field of health, speed and accuracy were taken as a priority. In this direction, the models considered to be worked on were revised by considering the data set and the target. As a result of the experiments, the targeted speed and 75% accuracy rate were obtained from the revised YOLOv4 model.

There may be many ways to increase the accuracy of the work. First of all, the size of the data set should be increased. Labelling can then be done more carefully by more than one dentist. This study can greatly speed up the diagnosis process of dentists in its current form. But to be more useful, the accuracy rate should be increased.

Diagnoses made by a dentist working in the oral diagnosis clinic in dentistry are divided into two. The first is the diagnosis of the patient's past. This study includes the classifications according to the first section. The next step is to diagnose the patient's prospective problems. If labelling can be done with more than one dentist in the future, our model can automatically detect all the tasks a doctor needs to do. It is possible that the study could be developed enough to enable computer detection of some dental diseases that a human might miss.

6. REFERENCES

Anonymous. *Colab*. Retrieved 25.05.2021 from https://cloud.google.com/gpu

Anonymous. *Darknet*. Retrieved 26.05.2021 from https://github.com/pjreddie/darknet

Anonymous. *Tensorflow*. Retrieved 26.05.2021 from https://www.tensorflow.org/

Apostolopoulos, I. D., Aznaouridis, S. I., & Tzani, M. A. (2020). Extracting Possibly Representative COVID-19 Biomarkers from X-ray Images with Deep Learning Approach and Image Data Related to Pulmonary Diseases. *Medical and Biological Engineering (2020)*.

Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv:2004.10934v1*.

Codella, N., Cai, J., Abedini, M., Garnavi, R., Halpern, A., & Smith, J. R. (2015). *Deep Learning, Sparse Coding, and SVM for Melanoma Recognition in Dermoscopy Images*.

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature, 542 (7639)*.

Farfade, S. S., Saberian, M. J., & Li, L.-J. (2015). Multi-view Face Detection Using Deep Convolutional Neural Networks. *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*.

Fuentes, A., Yoon, S., Kim, S. C., Unclaimed, & Park, D. S. (2017). A Robust Deep-Learning-Based Detector for Real-Time Tomato Plant Diseases and Pests Recognition. *Sensors 2017, 17(9)*.

Girshick, R. (2015). Fast r-cnn. *IEEE*.

HassanShibly, K., KumarDey, S., Tahzib-UlIslam, M., & MahbuburRahman, M. (2020). COVID faster R–CNN: A novel framework to Diagnose Novel Coronavirus Disease (COVID-19) in X-Ray images. *Informatics in Medicine Unlocked*.

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *IEEE*.

Lee, C.-S., Wang, M.-H., Yen, S.-J., Wei, T.-H., Wu, I.-C., Chou, P.-C., Chou, C.-H., Wang, M.-W., & Yan, T.-H. (2016). Human vs. Computer Go: Review and Prospect [Discussion Forum]. *IEEE*.

Lee, J.-H., Kim, D.-h., Jeong, S.-N., & Cho, S.-H. (2018). Diagnosis and prediction of periodontally compromised teeth using a deep learning-based convolutional neural network algorithm. *journal of periodontal & implant science*.

Li, X., Shang, M., Qin, H., & Chen, L. (2016). Fast accurate fish detection and recognition of underwater images with Fast R-CNN. *IEEE*.

Lin, T. (2018). *LabelIMG*. Retrieved 10.04.2021 from https://tzutalin.github.io/labelImg/

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). SSD: Single Shot MultiBox Detector. *ECCV*.

Maddison, C. J., Huang, A., Sutskever, I., & Silver, D. (2015). MOVE EVALUATION IN GO USING DEEP CONVOLUTIONAL NEURAL NETWORKS. *ICLR*.

MesutToğaçar, BurhanErgen, & ZaferCömert. (2020). COVID-19 detection using deep learning models to exploit Social Mimic Optimization and structured chest X-ray images using fuzzy color and stacking approaches. *Computers in Biology and Medicine*.

Motoki, K., Mahdi, F. P., Yagi, N., Nii, M., & Kobashi, S. (2020). Automatic Teeth Recognition Method from Dental Panoramic Images Using Faster R-CNN and Prior Knowledge Model. *IEEE*.

Ozturk, T., Talo, M., Yildirim, E., Baloglu, U., yildiirm, o., U.Rajendra, & acharya. (2020). Automated detection of COVID-19 cases using deep neural networks with X-ray images. *ELSEVIER*.

Prajapati, S. A., Nagaraj, R., & Mitra, S. (2017). Classification of dental diseases using CNN and transfer learning. *IEEE*.

Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., Ding, D., Bagul, A., Langlotz, C., Shpanskaya, K., Lungren, M. P., & Ng, A. Y. (2017). CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning. *arxiv*.

Raza, M., & Cinquegrana, P. (2018). *qubole*. https://www.qubole.com/blog/deep-learning-the-latest-trend-in-ai-and-ml/

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *IEEE*.

Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. *arxiv*.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., C, A., & Fei-Fei, B. a. L. (2015). ImageNet Large Scale Visual Recognition Challenge. *IJCV*.

Shen, J., Liu, N., Sun, H., Tao, X., & Li, Q. (2019). Vehicle Detection in Aerial Images Based on Hyper Feature Map in Deep Convolutional Network. *TIIS*.

Takahashi, T., Nozaki, K., Gonda, T., & Mameno, T. (2021). Deep learning-based detection of dental prostheses and restorations. *Sci Rep 11*.

Wang, S., Tang, C., Sun, J., & Zhang, Y.-D. (2019). Cerebral Micro-Bleeding Detection Based on Densely Connected Neural Network. *Frontiers in neuroscience*.

Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., & Summers, R. (2017). Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. *IEEE*.

RESUME

KAAN KÜÇÜK

EDUCATION

Master of Science 2017-2021	Akdeniz University Institute of Natural and Applied Sciences, Computer Engineering, Antalya, Turkey
Bachelor of Science 2009-2015	Gediz University Faculty of Engineering, Computer Engineering, İzmir Turkey