E-Dentist - Deep Learning in Detecting Teeth Conditions

Introduction

In recent years, deep learning has revolutionized the field of medical imaging, offering advanced techniques to diagnose and detect various health conditions with high accuracy. One of the most significant applications of deep learning in healthcare is its use in dental diagnostics. This proposal highlights the state-of-the-art algorithms in object detection—CNN, R-CNN, Fast R-CNN, Faster R-CNN, SSD, and YOLO—and compares these algorithms based on their use cases, accuracy, and efficiency in detecting dental conditions.

Neural Networks in Object Detection

Neural networks, particularly convolutional neural networks (CNNs), have proven to be suitable solutions for complex tasks requiring machine learning, including object detection. These algorithms are designed to identify and classify objects within an image, making them ideal for applications in dental diagnostics.

CNN and its Variants

Feed Forward Neural Network

It can be used to classify images to one label. However, the issue is with the number of inputs, for an image of size 100*100 with 3 channels, the input is 30000 possible pixels which is not suitable for a neural network.

CNN

CNNs reduce the number of pixels by focusing on important features through filters and pooling techniques, making them more efficient for image classification tasks.

CNN in Object Detection

CNN performs well in classifying an image. However, it can only classify an image to one label, in case of having many objects, it will label the image as one of them. Besides that, CNN uses sliding windows to capture features. For an image of size 600 and a filter size of 20, the total number of windows the CNN is going to use to capture the features is around **337, 000 windows** which consumes a lot of resources. The generated features would then pass through a Feed Forward Neural Network to classify the image.

R-CNN

Reduces the number of regions that need to be considered, from potentially hundreds of thousands (e.g., 300,000) to a much smaller number (e.g., 2,000) using a selective search algorithm **Region Proposal**, the output of selective search is a set of candidate regions (bounding boxes) that are likely to contain objects. The CNN would only process these generated boxes to extract features and then classify the regions to determine whether they contain objects and, if so, what those objects are.

R-CNN in Object Detection
Each generated box is cropped and re-sized to a fixed size suitable for CNN (a pre-trained CNN usually VGG16). The CNN acts as a feature extractor that generates a feature vector, each vector goes into two steps:
Object Classification: A fully connected neural network or SVM, classifies the features into object categories.
Bounding Box Regression: A regression model is used to resize and adjust the boxes.

Disadvantages
Each generated region proposal (around 2000) is processed independently through the CNN, and so the training time is slow.
Redundant Computation: Many regions would overlap, leading to redundant computation as similar regions are processed independently.
This makes real-time implementation impossible as each test image would take close to 47 seconds for execution.

Fast R-CNN
The main purpose of CNN as stated in R-CNN is to generate features. So Fast R-CNN would first process the whole image once through a pre-trained CNN to create a feature map, then region proposals are generated from Region of Interest (ROI) then ROI pooling is applied to extract fixed-size feature maps. Each region is then processed through two fully connected layers, one to classify the region and another to adjust the bounding box.

Improving Fast R-CNN
As stated in the Fast R-CNN paper, several optimizations can enhance the efficiency and performance of the algorithm. Below are the key improvements:

Mini-Batch Size: The paper suggests using a mini-batch size of 2, meaning that two images are fed to the CNN at each training iteration. After the convolutional features are generated, the Region of Interest (RoI) pooling layer selects 64 proposed regions per image, rather than the initial 2,000 regions. This reduction in the number of regions significantly improves computational efficiency.

Multi-Scale Training: To support images of different scales, image pyramids are used to generate four versions of each image at different scales. During training, one of these images is randomly selected for training, while the remaining three are used for validation. This approach helps the model generalize better to different scales.

Truncated Singular Value Decomposition (SVD): To further accelerate the training process, the paper recommends applying Truncated SVD to the weight matrices. SVD captures the most important weights and discards those that have minimal impact on the training data. Without SVD, 45% of the training time is spent adjusting the weights. With SVD, this is reduced to only 18%, leading to significant speed improvements.

This algorithm shows a significant reduction in time required for both training and testing when compared to R-CNN

Disadvantages
Fast R-CNN requires multiple stages of training. First, the network needs to be trained for classification and bounding box regression, and then selective search is used to generate proposals which are again fed into the network. The selective search is a separate algorithm, this separation can lead to suboptimal performance because the proposals are not influenced by the features learned by the CNN.

Faster R-CNN
Introduces the Region Proposal Network (RPN), which shares the same convolutional layers as the detection network. The RPN is trained to generate high-quality region proposals directly from the feature maps produced by the CNN which completely eliminates the selective search used in R-CNN.

How does RPN work? As stated in the paper, the algorithm defines 9 anchor points uniformly distributed around the generated features maps.
Then for each anchor point, 5 anchor boxes are generated at different scales and aspect ratios. These boxes act as initial guesses for potential object regions.
Then as always, the Faster R-CNN would go in two states, one to classify each anchor box as either an object, background, or a mix of them. The second state would adjust the offset of these boxes.

Faster R-CNN offers an improvement over its predecessors so significant that it is now capable of being implemented for real-time object detection.

All R-CNN family share
Using a CNN to extract features to classify objects and generate region proposals either before CNN or after it.
Re-sizing regions and normalizing them to be suitable for the fully connected layer, using insertion over union to label and train the region proposals, and using insertion over union to adjust the boxes.

Note
You need to label each object in the data using software by drawing a box around the objects with the type of the object,
This is necessary for interest of region (IoR) algorithm to be able to classify the objects and adjust the estimated bounding boxes. Note that the number of boxes generated by Faster R-CNN are lesser than those generated in R-CNN and Fast R-CNN, with the help of Insertion over Union, the proposed boxes are either eliminated as they capture a background (not an object) or merged together.

Issues with sliding windows (R-CNN Family)
The number of boxes needs to be large to capture the objects
The size of a box might be smaller or larger than the actual size of an object. A stride (sliding window step) might miss important contents of objects. Multiple phases are needed to train the data (classifier and region proposal). The network is slow when dealing with non-training data

New better way of detecting
One Stage Detector Algorithms

Single Shot Detector (SSD)
Consists of three stages

Single Shot: SSD performs object localization (predicting bounding boxes) and classification (assigning object classes) in a single forward pass of the network. This makes it faster compared to two-stage detectors like Faster R-CNN, which have separate region proposal and detection stages.

MultiBox: SSD needs to find where objects are in an image, It uses small filters (1x1) to make the process quicker and more efficient to predict box location. The only locations (boxes) considered are the ones with (IoU) with the ground truth label over 0.5, which is better than starting the prediction at random.

SSD uses pre-trained CNN VGG-16 because of its strong performance in high-quality image classification tasks. SSD removes the last fully connected layer and replaces it with a new layer that is able to detect multiple objects at different scales.

Training
You need to draw boxes around objects and label them, it's better to have different scales and ration of these boxes. It's recommended to use 6 different boxes for each part of the image.

YOLO
Uses a pre-trained model called DarkNet-19 consists of 30 Conveuolutional layers, 11 layers are used to detect the objects.
Yolo starts by dividing the image into cells (7*7 as mentioned in the paper) each cell predicts a fixed number of bounding boxes (5 boxes)
Each box contains a vector that has the coordinates of the object in the cell, the type of the object, and whether it's an object or a background.
Yolo uses **KMeans** to combine similar boxes together, leading to a reduction in the number of unneeded boxes.

The boxes can be adjusted through learning time, using (IOU) to classify and fix the labels.

One drawback of YOLO1 and SSD is that they sometimes fail to capture small objects. However recent developments in both algorithms have improved them a lot.

Preparing Data

Data Augmentation

This is the process of artificially increasing the size and variability of the training dataset by applying various transformations to the original images. This helps the model generalize better to new, unseen data.

Patches with Different IoU Ratios

IoU (Intersection over Union): IoU is a metric used to evaluate how well two bounding boxes overlap. It is the ratio of the intersection area to the union area of two boxes.

Patches

The authors generate additional training examples by taking patches (sub-images) of the original images. These patches are created with different IoU ratios relative to the original bounding boxes. For example, a patch with an IoU ratio of 0.1 means that the patch overlaps very little with the original bounding box, while an IoU of 0.5 means the patch overlaps halfway.

Purpose: By training the model on patches with different IoU ratios, the model learns to recognize objects at various scales and positions, improving its ability to detect objects in different parts of the image and at different sizes.

Random Patches

These patches might not necessarily overlap with any object but are used to increase the variability of the training data.

Horizontal Flipping

Each image is randomly flipped horizontally with a probability of 0.5. This means that for each image, there is a 50% chance that it will be mirrored along the vertical axis.

Purpose: This ensures that the model sees objects in both left and right orientations with equal likelihood. This is important because objects can appear on either side of an image in real-world scenarios.

Case Study

The paper ‘Comparative analysis of deep learning image detection algorithms make a comparative analysis of SSD, Faster-RCNN, and YOLO’

evaluates these algorithms using the Microsoft COCO dataset, which is a widely used benchmark for object detection tasks. The COCO dataset contains images with diverse objects in various contexts, making it suitable for assessing the performance of object detection models. The primary metrics used for evaluation are Average Precision (AP) and F1 score, which consider precision, recall, and Intersection over Union (IoU) between predicted and ground truth bounding boxes

First R-CNN
in the R-CNN model, there is separate training for the deep convolution network for feature isolation and the support vector machines for categorization. In the fast R-CNN method they have combined feature extraction with classification into a classification framework. The training time is **nine times faster** in Fast R-CNN than in R-CNN.

In the faster R-CNN method, the proposal isolation region and a bit of Fast R-CNN are put into a network template referred to as a region proposal network (RPN). The accuracy of Fast R-CNN and Faster R-CNN is the same.
All of these are built on top of **VGG** a pre-trained model.

Results comparison Discussion Following were some limitations that were observed in the three models SSD When it comes to smaller objects, SSD's performance is much worse as compared to Faster R-CNN. The main reason for this drawback is that in SSD, higher-resolution layers are responsible for detecting small objects. However, these layers are less useful

Conclusion
The conclusion of the paper highlights that YOLOv3 is the fastest algorithm among the three, making it suitable for real-time applications such as live video feed analysis. SSD is noted for providing a good balance between speed and accuracy, while Faster R-CNN, though the slowest, is recommended for applications where accuracy is more critical than speed. Overall, YOLOv3 shows the best performance in terms of speed, and its active development by the open-source community further enhances its capabilities.

Note:
Even though the above paper is old, still YOLO outperforms other algorithms in terms of speed and accuracy.

Recent searches have improved YOLO further, for example, YOLO5, YOLO7, and YOLO 8 include additional improvements in network architecture and training processes, and they introduce better algorithms for anchor box clustering and detecting small objects.

YOLO usage in a similar case
the paper titled "Automated detection and classification of dental caries using deep learning and panoramic dental radiographs" explores various algorithms and methodologies to enhance the detection and classification of dental caries in panoramic dental radiographs

YOLOv7: YOLOv7 is employed to detect both teeth and prostheses (inlays, crowns, implants, and bridges)

Optimization Algorithm
After the initial detection with YOLOv7, the paper implements a candidate optimization algorithm. This algorithm uses prior knowledge of dental anatomy to refine the candidate teeth detected by YOLOv7. The optimization focuses on evaluating the relative position and confidence scores of the candidates.

Stochastic Gradient Descent (SGD)
This was used during the training of the CNN-based object detectors, with specific values set for momentum and weight decay.

Image Augmentation Techniques
Techniques such as Mosaic, Translate, Scale, and Mixup were employed to reduce overfitting and enhance the training data.

Result
The mean Average Precision (mAP) for prosthesis categories was 97.36%, with a perfect precision value for bridge detection.

2nd case study
'Deep Learning Application in Dental Caries Detection Using Intraoral Photos Taken by Smartphones'

Case
The study aimed to apply deep learning algorithms to diagnose stages of smooth surface caries using images taken by smartphones. The researchers used a dataset consisting of 1,902 photos of the smooth surface of teeth taken with an iPhone 7 from 695 individuals.

Methodology
Four different deep-learning models were tested

Faster R-CNN, YOLOv3, RetinaNet, SSD

Results
YOLOv3 and Faster R-CNN showed the highest sensitivity among the four models for detecting cavitated caries.
YOLOv3 had a sensitivity of 87.4% for cavitated caries.
Faster R-CNN had a sensitivity of 71.4% for cavitated caries.

For visually non-cavitated (VNC) caries, the sensitivity levels were lower: 36.9% for YOLOv3 and 26% for Faster R-CNN.
The specificity of the models was above 86% for cavitated caries and above 71% for VNC.

Another paper titled Dental Cavity Detection Using YOLO
Using Recent YOLO alone achieved an accuracy of 87%

Final Thoughts:
YOLO stands out as the most promising algorithm for detecting small objects, particularly when dealing with small datasets. This effectiveness stems from YOLO's unique ability to perform object detection in a single forward pass, which allows for real-time detection with high accuracy. All the mentioned algorithms, including SSD and R-CNN variants, leverage pre-trained models. These models use backbone architectures like VGG, ResNet, or DarkNet, which are off-the-shelf algorithms already trained on extensive datasets like Microsoft COCO.

Data Labeling:
For these models to perform well on specific tasks, the data needs to be properly labeled. Accurate labeling involves marking the bounding boxes around objects and classifying them correctly.

Labeling Tools
There are several tools available for data labeling, such as LabelImg, RectLabel, and VGG Image Annotator (VIA). These tools facilitate the creation of labeled datasets by providing an intuitive interface for drawing bounding boxes and assigning class labels.

Data Labeling Process:
Collect Images: Gather a diverse set of images that represent the objects you want to detect.

Conclusion
Leveraging pre-trained models of algorithms like YOLO and maybe faster R-CNN can significantly enhance the performance of object detection tasks, especially when dealing with small datasets. Properly labeled data is essential for training these models, and various tools are available to assist in the data labeling process. Utilizing these resources, along with transfer learning techniques, can help achieve high accuracy in specific applications such as dental diagnostics.

Resources:

YOLO, RCNN family:

RCNN | Region-based | Selective Search | computer vision شرح عربي (youtube.com)

Fast R-CNN: Everything you need to know from the paper (youtube.com)

SSD:

Understanding SSD MultiBox — Real-Time Object Detection In Deep Learning | by Eddie Forson | Towards Data Science

Comparative analysis of deep learning:

Comparative analysis of deep learning image detection algorithms | Journal of Big Data | Full Text (springeropen.com)

Fast RCNN:

Fast R-CNN | IEEE Conference Publication | IEEE Xplore

Automated detection and classification of dental caries using deep learning and panoramic dental radiographs

Deep Learning for Caries Detection and Classification - PMC (nih.gov)

Deep Learning Application in Dental Caries Detection Using Intraoral Photos Taken by Smartphones

Applied Sciences | Free Full-Text | Deep Learning Application in Dental Caries Detection Using Intraoral Photos Taken by Smartphones (mdpi.com)