Segment and Lesion detection on dental OPG using Mask R-CNN

Niha Adnan, Owais Hussain

AKUH, KIET

Author Notes

Owais Hussain, Karachi Institute of Economics and Technology

Include any grant/funding information and a complete correspondence address.

**Abstract**

Abstract goes here

*Keywords*: machine learning, opg

**Experiment**

We developed a Multi-model[[1]](#footnote-0) deep neural network solution, using Region-based Convolutional Neural Network architecture. Two separate models were trained to achieve the target, i.e. identification of tooth segments in dental orthopantomograms (OPG), draw a mask around each segment, and spot the lesions in the OPG’s, if any existed.

The research community in the underlying domain has used several models for instance segmentation, including U-Net, DeepLab V3, Attention U-Net and V-Net (Volumetric). While segmentation applications have been seen adopting U-Net and its variations due to its performance on smaller datasets, we opted against it in favour of Mask R-CNN1 due to its adoptability for transfer learning. Therefore, instead of learning our model from ground up, we loaded weights from a Mask R-CNN model, pre-trained on COCO dataset2, therefore leading us to train the model in much fewer iterations. Mask R-CNN is an extension of Faster R-CNN3 and was particularly selected because of its performance in tasks that require precise object segmentation. For example:

1. He4 compared Mask R-CNN with other state-of-the-art object detection algorithms on COCO dataset and concluded that Mask R-CNN outperformed other algorithms in image segmentation, which requires accurate delineation of object boundaries.
2. Huimin5 chose Mask R-CNN as the base model for 3D segmentation from CT images. Their results show that Mask R-CNN outperformed Faster R-CNN in accuracy and segmentation quality.

The dataset used in this study was collected from three different sources and contained a total of 230 OPGs. We divided the dataset into training, validation, and test sets, consisting of 203, 24, and 3 OPGs respectively.

## Model Training

Models were trained on a modest computer equipped with Nvidia GTX-1060, a CUDA-based GPU, on Tensorflow 2.0[[2]](#footnote-1), paired with Intel’s 8th Generation Core i7 CPU and 16GB DDR4 RAM.

The backbone for Mask R-CNN used is ResNet[[3]](#footnote-2), a deep neural network architecture for training convolutional neural networks (CNNs). In particular, the experiments used ResNet-50, which has 50 layers. The anchor scales, which determine the size and aspect ratio of the anchor boxes, were chosen to be of sizes 8, 16, 32, 64, and 128. Moreover, 32 RoI (Region of Interest) samples were drawn for each image during training. The initial weights for the model were loaded from Coco model2. The training process was run continuously, until the marginal improvement between epochs – determined by epoch loss – dipped below the 1% threshold.

We trained the tooth segmentation model for 310 epochs, with each epoch taking 100 iterations. The training process took 12 hours, with an average of 135.4 seconds per epoch. The batch size for training the model was 2, and the images were scaled to maximum Width and Height of 832 pixels. Each tooth was identified by a unique label, or class\_id. Therefore 32 labels were used: 11-18; 21-28; 31-38; 41-48 according to the position on 4-quadrant grid. After training each epoch, the results were stored as a model snapshot.

Similarly, the lesion detection model was trained for 500 epochs, each running for 100 iterations. This model took 11 hours of training, with an average of 63.8 seconds per epoch. The batch size for this model was 1, while the other parameters remain similar to that of segmentation and masking model. In addition, the data was augmented for this model to include variety as well as expand the number of examples. The augmentation techniques applied are:

1. Cropping and padding each side of the lesion polygons by 25%.
2. Blend in Alpha channel and remove about 50% of the colour.
3. Flip random 50% of the images horizontally and random 50% vertically.
4. Scale images from 75% to 125% of their original size
5. Rotate images between -30 and 30 degrees
6. Sheer images by -15 to 15 degrees

Once trained, both models were sequentially run to predict tooth segments, as well as lesion areas on same test images. The sequence of actions on each test case is as follows:

1. The target OPG is passed to the segmentation model.
   1. Tooth segments are classified by the model, and are assigned class IDs (labels).
   2. A mask is predicted for each tooth segment, defined as a polygon.
   3. A bounded box is created around the predicted tooth segment.
   4. The predicted labels and their corresponding Mask polygons are saved in a file.
2. The target OPG is passed to the lesion detection model.
   1. The model spots lesions on the entire image at once and defines a polygon for each lesion.
   2. A bounded box is created around the predicted lesion.
   3. The predictions are saved in a file.
3. The output from segment model is drawn on top of target OPG such that each tooth segment is labeled with a respective classification score; the masks are painted over each tooth segment with a different predefined colour; a bounded box of same colour is drawn around the mask, as shown in Figure [? to ?].
4. The output from lesion model is drawn on top of target OPG such that each lesion is separately painted with a distinctive colour, along with its classification score; a bounded box is drawn around the lesion area, as shown in Figure [? to ?].
5. Both the outputs are superimposed for a combined feedback, such that the segmentation model output is overlapped on lesion model output with 50% transparency.

## Evaluation

The performance and accuracy for the individual models were computed on 6 metrics commonly used to evaluate Mask R-CNN models.

Epoch loss (): the loss function for Mask R-CNN model. It is the weighted sum of several sub-losses, each of which measures the difference between the predicted values and the ground-truth values. It does not have a fixed upper limit but, its lower bound is 0. Epoch loss is given as

1. Class loss (): tells how well the model recognize each class of object. It is given as the average binary cross-entropy loss over all positive RoI samples for each tooth label (), specifically, for each positive RoI. In this definition of multi-class loss allows the network to generate a mask for each class independently, i.e., without competing with each other. Algorithmically, if the ground-truth class label for the RoI is , then the loss is the binary cross-entropy between the predicted class probability for and its ground-truth label. Therefore, for all other class labels , the loss is the binary cross-entropy of probability for predicted class . All of the positive losses for each class are then averaged to calculate .

Bounded box loss (): measure how well the model localizes each object detected. It averages bounded box losses for each classification. It is calculated as smooth loss (also called Huber loss) between predicted bounded box coordinates and the ground truth for each positive RoI. Algorithmically, if the ground truth label is a background class, then the loss is set to 0, otherwise it is calculated as the squared term loss between ground truth and predicted box for each class.

1. Mask loss (): measures how well the network model segments objects which were labelled. Similar to , it is also given as the average binary cross-entropy loss. The predicted mask is generated using a binary mask head of size 28x28, applied to feature map of each RoI proposed. This mask head is then reshaped to the size of the input RoI. Finally, the loss calculated between the ground-truth mask and the predicted mask for each class instance. Therefore, if is the number of RoI’s, is the ground truth mask for RoI, is the predicted mask, then the loss is calculated as:
2. R-PN class loss (): measures the loss similar to for the backbone ResNet-50 Region Proposal Network.
3. R-PN bounded box loss (): measures the loss similar to for the backbone ResNet-50 Region Proposal Network.
4. Accuracy: calculated as where is True-Positives, or the number of segments predicted correctly; is False-Positives, or the number of segments predicted incorrectly; is False-Negatives, or the number of segments not predicted.
5. Precision: tells when a model classifies a segment, how correct the class label is. It is calculated as
6. Recall: tells out of all segments which existed, how many were retrieved by the model. It is computed as
7. F1-score: this is the harmonic mean of Precision and recall, and gives a balanced measurement between the two. It is given as
8. Average prediction score: the probability with which the model assigns label to classified segments. This is calculated using the Softmax function applied to the output of final fully connected layer, which produces a probability distribution over all possible classes.

Table 1 lists the values for each of the evaluation parameters for both the segment and lesion detection models.

*Table 1:Evaluation metrics of Mask R-CNN models*

| **Metric** | **Segment Model** | **Lesion Model** |
| --- | --- | --- |
| epoch\_loss | 0.29095 | 0.88055 |
| mrcnn\_bbox\_loss | 0.03662 | 0.22071 |
| mrcnn\_class\_loss | 0.08350 | 0.08456 |
| mrcnn\_mask\_loss | 0.12950 | 0.26485 |
| rpn\_bbox\_loss | 0.03893 | 0.29863 |
| rpn\_class\_loss | 0.00240 | 0.01180 |
| accuracy | 0.54805 | 0.46667 |
| precision | 1.00000 | 0.80556 |
| recall | 0.54805 | 0.55000 |
| f1\_score | 0.70610 | 0.63095 |

## Model progressions over time

Aside from the overall evaluation, we used Tensorboard utility, provided by TensorFlow toolkit, to measure how the models progressed over time. The graphs [REF] report each type of loss during training.

| **Segment Model** |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |

| **Lesion Model** |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |

Predictions on test images:

TODO:



1. He, K., Gkioxari, G., Dollar, P. & Girshick, R. Mask R-CNN. in *Proceedings of the IEEE international conference on computer vision* 2961--2969 (2017).

2. Lin, T. Y. *et al.* Microsoft COCO: Common objects in context. in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* vol. 8693 LNCS 740–755 (2014).

3. Girshick, R. Fast R-CNN. in *Proceedings of the IEEE International Conference on Computer Vision* vol. 2015 Inter 1440–1448 (2015).

4. He, K., Girshick, R. & Dollar, P. Rethinking imageNet pre-training. in *Proceedings of the IEEE International Conference on Computer Vision* 4917–4926 (2019). doi:10.1109/ICCV.2019.00502.

5. Cui, Z., Li, C. & Wang, W. Toothnet: Automatic tooth instance segmentation and identification from cone beam ct images. in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* vols 2019-June 6361–6370 (2019).

1. As opposed to Ensemble learning, where individual models are trained on different data, algorithms, and parameters to be combined later to improve overall accuracy and robustness, Multi-model learning trains multiple models independently on the same dataset to learn different representations and capture complementary information. [↑](#footnote-ref-0)
2. An open-source machine learning framework developed by Google to build and train machine learning models. TensorFlow 2.0 is the latest major version, which was first released in 2015. [↑](#footnote-ref-1)
3. “Residual Network”, which uses of residual connections that allow for the creation of very deep networks without the problem of vanishing gradients. [↑](#footnote-ref-2)