Identification of Cavities In Dental Imaging

Nicholas C. McCormick & Alexander Bailey & Funminiyi Bamidele

Shiley-Marcos School of Engineering

AAI – 541: Team 5

Edwin Addison, JD, MBA

August 14th, 2023

Abstract

Dental caries (cavities) have become a common part of life for adolescents and adults in today's society. According to the Center for Disease Control and Prevention, in children 6 to 8 years of age over half (52%) have had a cavity in their baby teeth. The rate climbs higher in adolescents aged 12 to 19 with 57% having at least one cavity in their adult teeth. And lastly, adults aged 20 and older, 90% have had at least one cavity (Cavities, 2023). With the soaring cost of dental care, it has become common for a single cavity filling to cost well upwards of $200 (Humana, 2023). Current practice for identifying cavities includes two methods. X-ray imaging of teeth using specialized and costly equipment. These x-rays then still need to be scrutinized by a dentist. Secondly, a dentist can look at the teeth directly for signs of caries. Dentists currently miss identifying 50% of dental decay and wrongly diagnose 30% (Rudy, 2023). This research used both colored dental images and x-ray images to train an ai algorithm based on a pre-trained Faster Region-based Convolutional Neural Network (R-CNN). The goal of the research is to make a machine learning algorithm capable of detecting dental caries in x-ray and colored images with an accuracy greater than that of a Dentist. The final model was able to accurately identify cavities with a precision score of 0.5505, average precision of 0.5610, recall of 0.9478, and F1 Score of 0.6965. The mean Average Precision (mAP) of the model was 0.3846. Recall represents the number of true positive objects detected, while precision represents how many of the predicted positive detections were actually correct. mAP is the mean of average precisions across all classes in the model and best represents the effectiveness of the model. The results of this study show that using computer vision based deep neural networks can be an effective way to identify cavities with more model and data refinement.

Scope of the Project

The scope of this project was to utilize the effectiveness of machine learning in the space of computer vision in order to build a model that can reliably detect dental caries from x-ray images, and photographs of teeth. The project took place over the course of four weeks. During that time, we sourced, explored, and cleaned the datasett. Next, a pretrained model architecture was chosen and applied to the data set for training and testing. Next evaluation of the model's effectiveness took place. Lastly, a report of the research findings was created along with a presentation outlining the results and effective uses of the research.

Data Sources and How They Were Prepared

The data used for this project consist of 1,090 images with a size of 640 x 640 pixels. 595 of the image dataset are categorized as x-ray images, while the remaining 445 images are categorized as color images. All the images were consolidated into one image set and provided by the authors of *"A hybrid mask RCNN-based tool to localize dental cavities from real-time mixed photographic images"* (Rashid et al., 2022). The data was then hosted on Zenodo by Umer Rashid. Metadata was also attached to each of the images that provided the annotations for identified cavities. What this essentially means is that a dental professional already went through the images and placed bounding boxes around every dental cavity they were able to identify. With only four weeks to complete the project it was pertinent that a data set that already contained identified cavities was selected. This was necessary because the alternative method for data collection would have been collecting dental images from a dentist's office, then getting one of the dentists to go through all the images and annotate them for cavities manually. This would have been a quite costly and time-consuming endeavor that would likely not have been possible within the four-week time frame.

Next, the annotation metadata needed to be loaded in and properly attached to each of the images. The annotation metadata was converted to a single JSON file with all of the annotations kept in COCO format. COCO stands for Common Objects in Context and is a widely used dataset format utilized for labeling objects. This is also beneficial in a production environment because it allows for the addition of more data in a uniform way and standardizes the annotation process. The annotations had the keys, info, licenses, categories, and annotations. A PyTorch compatible dataloader was created and implemented to fulfill this task. This was done by defining a custom data set loader from datasets.VisionDataset, with help from the cv2 library. The cv2 library is a computer vision library that utilizes python to perform operations on image files. Using VisionDataset allowed the creation of a label map, selection of a target image, target annotation, and ultimately the return of corresponding tuples for the given index.

Description of Solution Approach Overview

1.1 Exploratory Data Analysis As outlined above, a research topic was chosen; cavity detection using a Faster R-CNN model with a ResNet-50 backbone. Next a data set was found that could potentially offer insight into the chosen research topic. Then exploratory Data Analysis (EDA), was performed to understand the data set and figure out how to modify it. Exploratory data analysis included writing a function to view how many color images were present (445) and how many X-ray images were present (595). Next, verification of the COCO_data categories were performed. The annotation categories were shown to be 'dental', 'cavity', 'pockets', and 'tooth'. After establishing annotation categories the total number of each annotation was obtained. The label 'dental' had 0, 'cavity' had 2221, 'pockets' had 450, and 'tooth' had 2119 counts as seen in the graph below.

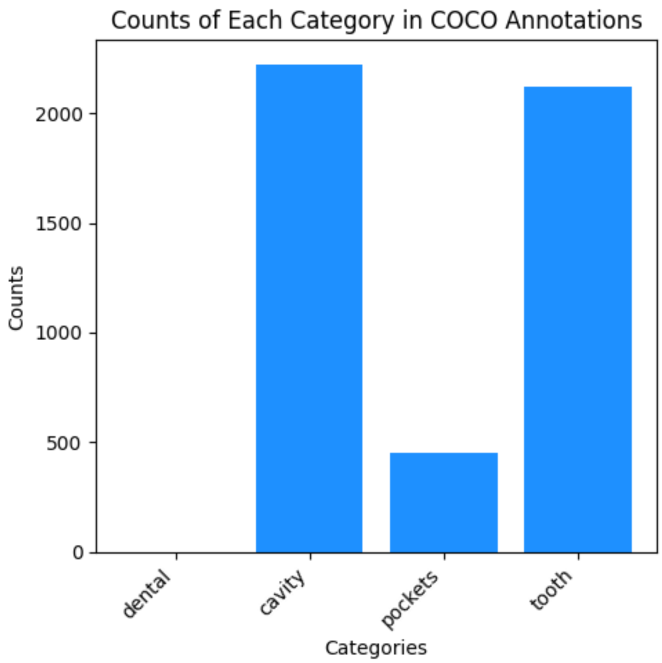

Figure 1 – Shows the counts of each category in COCO annotations.

In addition, visualization of the images with their corresponding annotations and bounding boxes

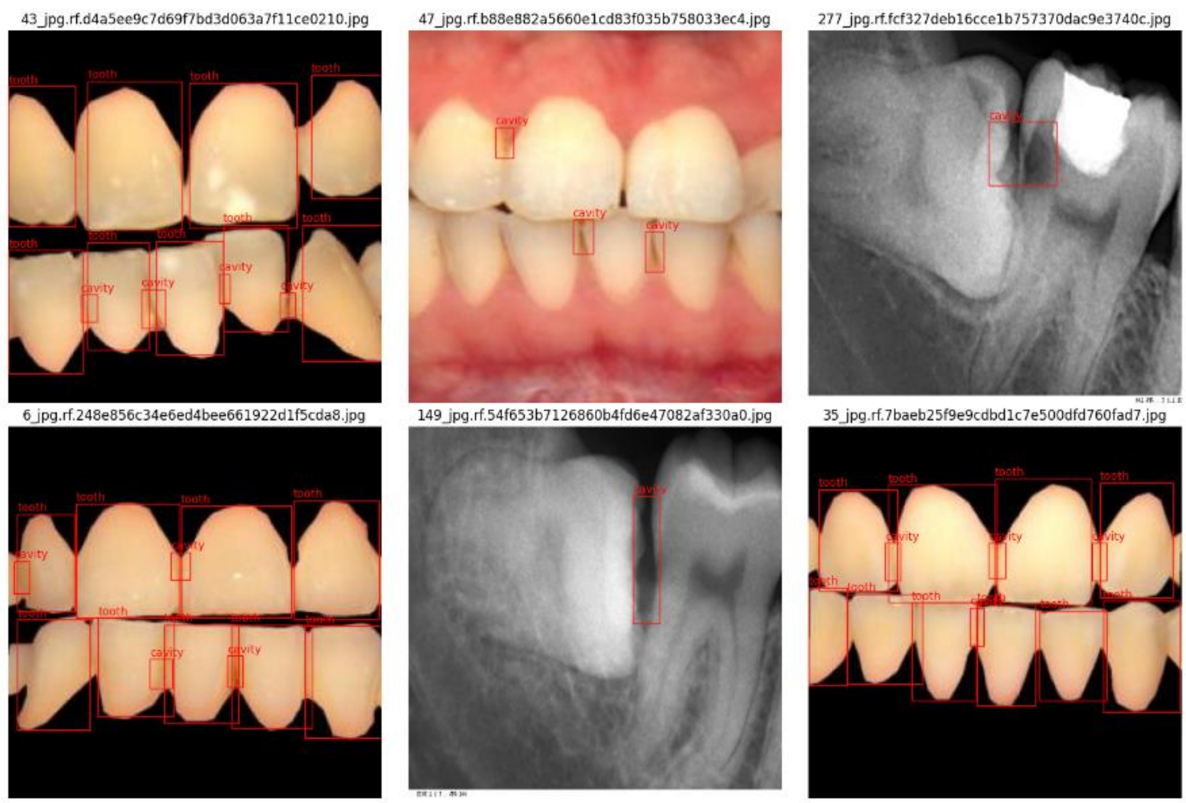

Figure 2 – Shows each represented COCO dictionary category and bounding box around the identified object.

The last step of EDA involved determining if any of the images did not contain any bounding boxes. This step was necessary to understand how the data would need to be modified. 27 images in total were found to not contain any annotated bounding boxes.

1.2 Data Preprocessing

After EDA was complete data preprocessing took place. The first step was creating and defining a custom PyTorch data set. In order to do so, DentalCavityDatasetV4 was created following the conventions explained in the Pytorch documentation for preparing a custom dataset to be compatible with their data loader function. This is necessary in order to load and preprocess the dataset for use in training and evaluation of the R-CNN object detection model. Within DentalCavityDatasetV4 an initializer was defined based on the data root, transforms, annotations and our new 'id's', which are just 'cavity' and 'non-cavity' since the scope of the project is focused on only the detection of cavities. The length of the data set (number of images) is also defined. In addition, __getitem__ is also defined. __getitem__ was responsible for acquiring and processing images and their corresponding annotations at a given index. __getitem also used 'PIL.Image' to open an image and convert to a NumPy array of floating-point values in range [0.1]. Additionally, __getitem__ parsed bounding boxes and labels from annotations associated with each image. If no bounding boxes were present in the COCO data of an image, a bounding box was created around the entirety of the image and labeled as 'non-cavity'. Lastly, __getitem__ converted bounding boxes and their labels to PyTorch tensors, creating a target dictionary where necessary transformations as defined in their functions are carried out. Lastly, a separate instance of DentalCavityDatasetV4 was initialized ('full_dataset'), and used to split the data into training, validation, and testing subsets prior to being loaded into the torchvision data loader.

Next, the transformation functions called by the DentalCavityDatasetV4 class needed to be defined. Since the image data set was rather small (less than 200 images) a way was needed to augment the data to achieve greater model accuracy and efficiency. After exploratory data analysis was completed, two methods for image transformation were selected.

The first method for image transformation was a combination of converting images to gray-scale and then randomly selecting images to flip over their respective horizontal and vertical axes. The second method was to only randomly select images and flip them at random over their horizontal and vertical axis. The first method was selected because of the efficiency associated with processing of grayscale images in a deep neural network. When a computer vision deep learning network is being utilized on color images it must perform convolutions on three distinct color arrays (red, green, blue). When converting to grayscale, values are stored in a single array, which means the computer vision model must only compute a single convolution (Nelson, 2020). Ultimately, if the model can work effectively on gray-scale it should be less computationally expensive and faster, leading to a more production friendly version. Despite being more efficient, using grayscale transformations on colored images can have adverse results. For example, the reduction of dimensionality through the loss of color-related information, spatial information, and color specific object identification can result in a less accurate model (Nelson, 2020). For this reason, two models were developed.

For the first model iteration a dataset transformation pipeline labeled 'transform_V1' was defined. This transformation converted the images to NumPy arrays from tensors, then converted the image to gray-scale while keeping all three RGB channels. Next, the 'transform_V1' function will flip the image over either the horizontal or vertical axis, each of which have a 50% probability of occurring. Lastly, each image is then converted back to a tensor with channels as the first dimension – [Channel, Height, Width].

For the second model iteration a dataset transformation pipeline labeled transform_V2 was defined. This transformation performs the same augmentations as transform_V1 but without the conversion to grayscale step so that pixel values remain in their original but normalized form.

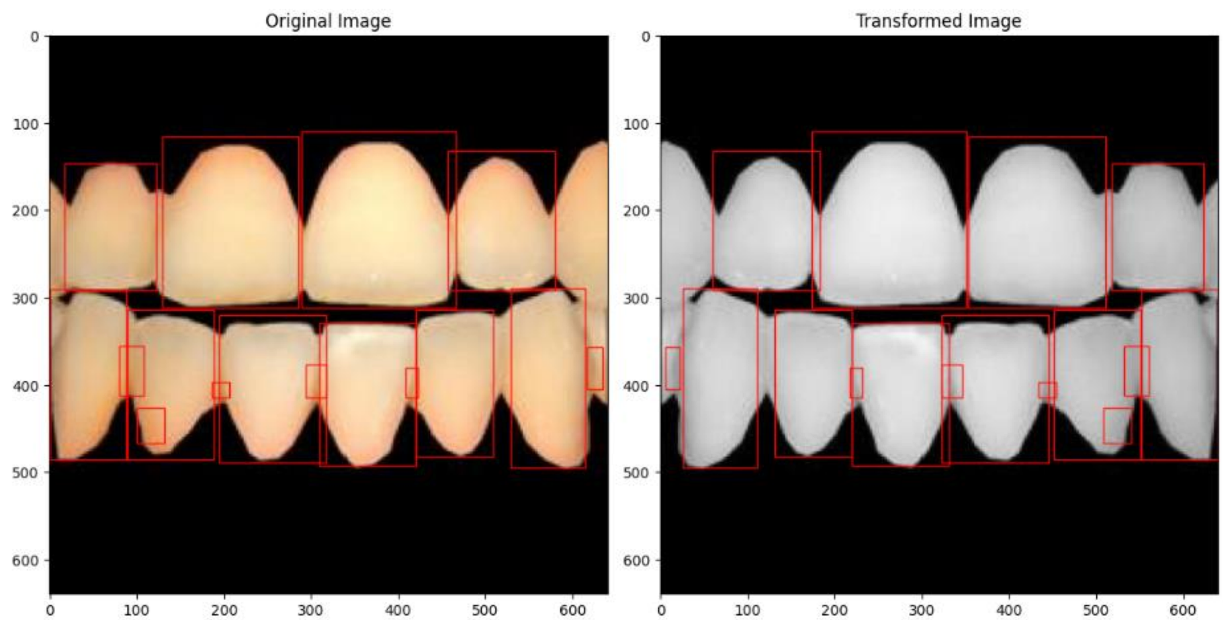

Figure 3 – The left image is the original before being transformed to grayscale which is seen in the image on the right.

Now the training loop and different model formats can be discussed.

Algorithms and Tools Chosen / Description of the Training Process

1.1 Faster R-CNN Selection Reasoning

For this specific project a faster R-CNN was chosen for a few different reasons. The faster R-CNN was designed to be a faster version of the R-CNN model. This is advantageous in the dental industry as it would allow for faster recognition of dental cavities and subsequently faster treatment of the cavities. For example, with higher accuracy and less false positives than a dentist, the model would be able to give a patient instantaneous feedback on how many cavities they have. This could increase their trust in the dentist because the dentist would now be much less likely to miss the detection of present cavities. Additionally, earlier and more accurate detection of cavities, could reduce the likelihood of cavities progressing to the point that they kill the tooth, which would then require a root canal. Root canals are significantly more costly and time consuming to the patient than a filling.

The faster R-CNN differs from a regular R-CNN in one major way. The faster R-CNN utilizes the integration of a Regional Proposal Network (RPN) into the original network whereas a standard R-CNN model is dependent on the use of external algorithms such as selective search (Gandhi, 2018). RPNs work by analyzing an input image for possible objects and creating object scores where bounding boxes would likely be created. The Faster R-CNN uses regions of interest pooling to extract features from the identified regions. Rather than having to process potentially thousands of region of interest (ROI) proposals individually per image, the Faster R-CNN shares its convolutional layer with the detection network allowing processing to be performed just once per image (Gandhi, 2018). The Faster R-CNN then picks a class label for each identified region and adjusts the bounding box associated with it to get a more accurate boundary as it learns (Gandhi, 2018).

The second reason a Faster R-CNN model was chosen was due to the research previously conducted on the topic of cavity identification in dental images. In the study conducted by Rashid et al., (where data for this project was gathered from as discussed previously) they used a hybrid mask RCNN-based tool. While discussing their results they stated regarding their model, “The evaluations performed by the dentists showed that the correctness of annotated datasets is up to 96%, and the accuracy of the proposed system is between 78% and 92%. Moreover, the system achieved the overall satisfaction level of dentists above 80%” (Rashid et al., 2022). The exceptional results shown in their study illustrated that an RCNN model would likely be a good choice to pursue for the purposes of this research. Ultimately for this study it was decided that the Faster R-CNN could prove to be a more cost effective and efficient alternative to the hybrid mask R-CNN used by Rashid et al.

1.2 Training, testing, and validation split

Before running the Faster R-CNN, the training loop needed to be defined. First, two new lists that store training and validation losses are created. The training loop function also takes into account the number of epochs, a learning rate scheduler, optimizer, and the object detection model trained. It also defined that for each epoch, the training loss would be tracked (epoch_train_loss), data would be processed through the training loader in batches where each batch is sent to a specified processing device (GPU) and converted to a float tensor. Each batch then has a forward pass performed, followed by a backward pass and optimization step to update the model parameters. The training loop then had a validation phase that operated similarly to the epoch loop in order to track the cumulative validation loss for each epoch.

After the training loop was set the data could then be split into our test, train, and validation sets. The data was split into 85% training, 5% validation, and 10% testing sets. This was done by utilizing a custom randomized split function to create three directories for training, validation, and testing. Then the training, validation, and test datasets were initialized via the transformation functions dependent on the inputs; the dataset directory, COCO formatted data, and the corresponding specific transformations. In addition, to make sure the data will be compatible with the Faster R-CNN model structure, a function was added to remove all images from the dataset that still do not have bounding boxes. After this, the data loaders for the training set, validation set, and test set were initialized. The data loaders were responsible for loading batches of data during training, validation, and testing. Lastly, a simple check is performed on the first batch to ensure no empty bounding boxes exist. At this point the train_dataset consisted of 884 samples, the validation dataset consisted of 52 samples, and the test_dataset consisted of 104 samples. The total number of images in the first batch was 16.

1.3 ResNet-50 & Faster R-CNN

Next, we loaded in the pre-trained Faster R-CNN model with a ResNet-50 backbone. The ResNet-50 backbone is useful because it helps solve the vanishing gradient problem that can occur in deep neural networks. ResNet-50 is essentially a pre-trained neural network architecture that was trained on an image data set called ImageNet which consisted of over a million varied images (ResNet-50: The Basics and a Quick Tutorial, 2023). Since the image data set was so limited, the pretrained ResNet-50 model helps boost the model accuracy, by supplementing the model with pretrained weights from training on the ImageNet database. The Faster R-CNN model is able to utilize the pretrained model to improve its accuracy. The Faster R-CNN model utilizes a structure called a Feature Pyramid Network (FPN) in partnership with the ResNet-50 backbone that can extract features and generate multiple feature maps at once that end up stacked in the shape of a pyramid because multiple image scales are being analyzed for features in tandem. The lower levels of the pyramid are capable of detecting larger objects, while higher levels are better at detecting smaller ones (Hui, 2018). It is advantageous when detecting objects of different sizes and excels at capturing small details in a larger context.

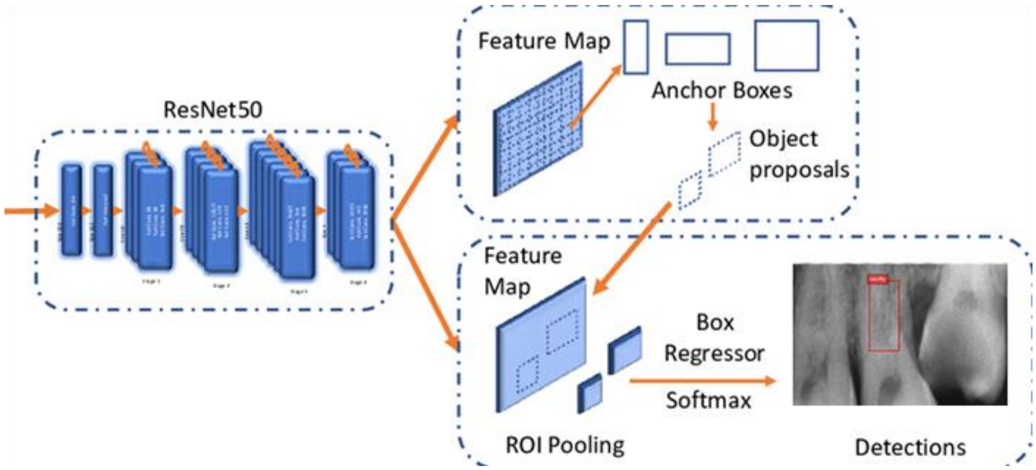

Figure 4: Faster-RCNN w/ Resnet50 Backbone (De et al., 2017)

1.4 Setting Model Parameters

The last step before training the models was identifying the number of input features in the classifier, defining our parameter, optimizer, and scheduler. The number of classes was set to 3 because the only three classes present at this point are 'background', 'cavity', and 'non_cavity'. The optimizer was set to Stochastic Gradient Descent (SGD). SGD was chosen because it adds controlled noise to the data set to avoid convergence and gives the model better generalization capabilities while also being extremely efficient with processing power and memory usage (Dutta, 2019). This is very beneficial to a production environment because it keeps the cost of processing data much lower. The learning rate was set to 0.005, momentum to 0.9 and a weight decay at 0.0005. Momentum is an accelerator for gradient vectors that speed up the time to convergence while weight decay is a regularization technique that prevents weights from getting too large to reduce overfitting. Training was conducted over 20 epochs with a batch size of 16. Training time took roughly 3 minutes per epoch using a V100-cloud based GPU provided by Google. Furthermore, a step-based scheduler was initially created with a step size of 3 and gamma of 0.5, however . This means that at every 3 epochs the learning rate was reduced by multiplying the learning rate by the gamma value. However, this scheduler was removed from the optimizer as it was found to hurt the learning performance of our model and contributed negligible performance increases when experimenting with other gamma values and step sizes. It was concluded that the scheduler would only be beneficial with a larger amount of epochs, but due to the length of training time required, it was limited the number of epochs to 20.

Results

Learning curves provide valuable insights into the performance and progress of machine learning models as they are trained over epochs. Learning curves are typically observed to determine a model's training and validation performance. A key aspect to analyze is whether there is a consistent gap between the training and validation curves. A large gap might indicate overfitting, where the model performs well on training data but poorly on new, unseen data (validation). Conversely, a narrow gap suggests a better generalization capability. We have three models, each representing a different iteration or version: "Base Model," "Model 2," and "Model 3." These learning curves reflect the evolution of these models' performance over training epochs as we tuned SGD hyperparameters and augmentation methods

Base Model **Model 2** **Model 3**

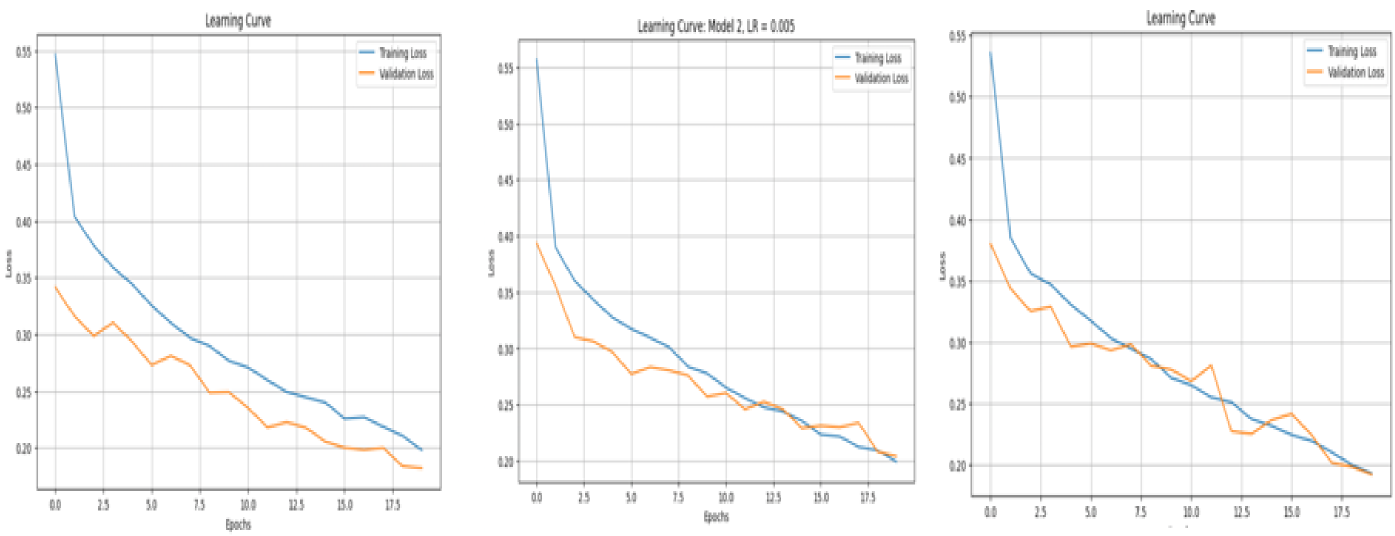

Figure 5: Comparison of training/validation loss curves for each model

1.1 Model Evaluation - Conclusion

The results obtained from the Faster R-CNN model's performance in dental cavity detection highlights the well-justified selection of this model architecture and provides insights into the implications of its findings. Part of the rationale behind choosing the Faster R-CNN model is its support by the prior research conducted in dental cavity identification. The study conducted by Rashid et al. 2022 offers remarkable insights into the potential of RCNN-based tools in dental imaging. Their findings underscore the high correctness of annotated datasets and the overall satisfaction level of dentists, indicating the viability of such models for cavity identification. In light of this compelling evidence, The model parameters settings, include the choice of Stochastic Gradient Descent (SGD) optimizer, learning rate, momentum, and weight decay, this represent our comprehensive approach to mitigating overfitting and ensuring efficient convergence. The inclusion of a learning rate scheduler further enhances the model's adaptability during training.The evaluation metrics, including Precision, Recall, F1 scores, and mean Average Precision (mAP), provide a comprehensive insight into the model's performance across multiple iterations. Mean Average precision is calculated by comparing the intersection over union (IOU) to measure the amount of overlap between the prediction and ground truth bounding boxes. The progressive improvement in metrics across "Base Model," "Model 2," and "Model 3" underscores the iterative refinement process.

This model evaluation revealed intriguing insights. The Precision-Recall trade-off was evident, with the "cavity" label demonstrating a precision of 0.4350 and recall of 0.9609, indicating a moderate capability in cavity detection. The "non_cavity" label showed a higher precision of 0.5040 and recall of 0.9960, excelling in non-cavity recognition.

The mean Average Precision (mAP) score of 0.3644 indicated a moderate overall model performance. Interestingly, the individual Average Precision (AP) scores per class varied, with "cavity" scoring 0.4714 and "non_cavity" achieving 0.6218. These results align with the model's specialization in these classes. It's important to note that while the model showcases notable potential, further refinement and exploration are possible to enhance its accuracy and performance across all classes. This evaluation underscores the significance of model selection, training methodology, and parameter tuning in achieving effective object detection for dental cavity identification. As can be seen by the sample prediction image below (Figure 8), the model was able to successfully identify cavity regions, however it did not always catch all occurrences of cavities.

	Precision	**Recall**	**F1**	**mAP**
Base Model	0.4350	0.9609	0.5989	0.4714
Model 2	0.4481	0.9565	0.6103	0.5477
Model 3	0.5505	0.9478	0.6965	0.5610

Figure 7: Comparison of each model’s evaluation results for detection of the ‘cavity’ class label

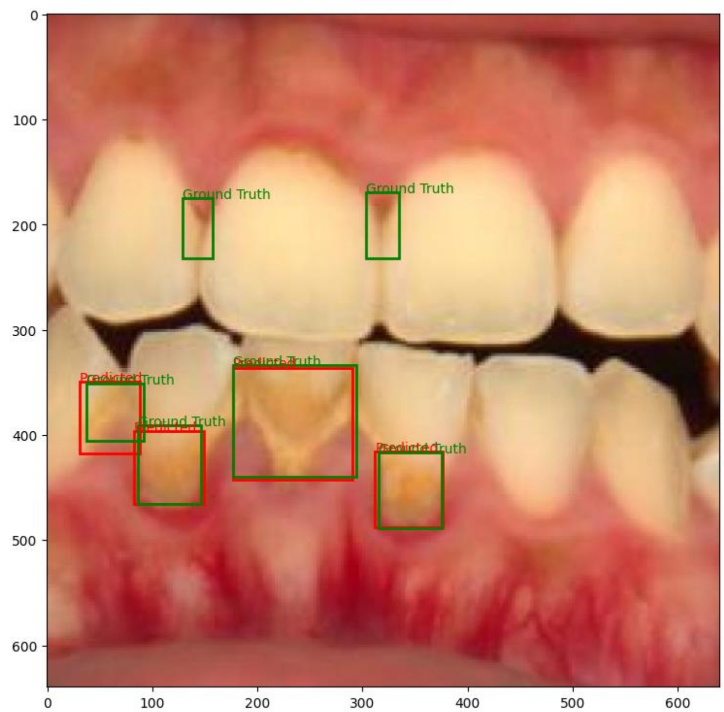

Figure 8: Prediction sample showing predicted values (red) and ground truth values (green)

In conclusion, the model demonstrates good performance in detecting cavity and non-cavity regions, thus showing capability of detecting dental caries in x-ray images and colored photos of peoples' teeth. The thorough exploration of the Faster R-CNN model's performance in dental cavity detection has yielded valuable insights into its effectiveness and potential applications. The selection of the Faster R-CNN model architecture was grounded in its ability to enhance the efficiency of dental cavity recognition.

Provided more time and resource was given to this project, a similarly designed model using Ultralytics YOLOv8 may have been constructed to compare results. In addition, more processing power and memory, as well as a larger dataset could have been utilized to increase the accuracy of the model. Despite this, the project successfully acted as a proof of concept by demonstrating the potential use of computer vision deep learning networks to increase the accuracy of dental cavity recognition.

Ethical Considerations

The healthcare industry is arguably one the industries that needs to be the most considerate when it comes to ethical concerns. Regarding patient data, x-ray images and photos of teeth provide enough information about a person to identify them. Everyone's teeth are unique to themselves and can be used in a similar manner to that of a fingerprint. The ethical concern related to this is that the identification of their teeth can be considered their own private identifiable information. If we were to implement our cavity detection model, all the data would need to be kept as secure as dental records currently are. This would be a relatively easy need to meet because most dental offices already contain digital records of patients' x-rays. All that is needed is the API of the current system already in use. The only other security addition would be the use of cloud computing for implementation of the algorithm (Data Protection and Privacy | AWS, 2023). Since AWS is such a widespread data processing and storage solution; they have built-in security features that are ready to use. More specifically, dental patient records would be subject to HIPAA privacy laws, AWS clearly lists on their website that companies subject to HIPAA actively currently use their services and that they meet a myriad of international security standards for cloud computing such as ISO 27017, ISO 27701, and ISO 27018 (Data Protection and Privacy | AWS, 2023). This makes using AWS cloud computing an easy to make choice for ethical security concerns.

The more pressing ethical issue is present when gathering data for the training of the model. The concern being that people may need to consent to their personal dental records being used for research. These permissions could take a long time to gather and would need to be continuously sought out by the dental practice on the researcher's behalf. A data sharing agreement between the research company and dental office would also need to be created. Lastly, this could possibly be avoided by only using the records for training and anonymizing the data so that no information about the patient whose teeth appear in the images is known.

The last identifiable ethical concern arises when discussing patient care. When discussing oral care, the Center for Disease Control and Prevention identified the following in adults aged 20 to 64 years old, "Nearly twice as many non-Hispanic Black or Mexican American adults have untreated cavities as non-Hispanic White adults" (Disparities in Oral Health, 2023). A disparity in oral healthcare for people younger than 20 and older than 64 is also clearly illustrated in the article. The disparity in racial or ethnic groups shows that considerations should be put in place for this new cavity detection software. It could provide more accurate screening to these individuals and reduce the number of visits to the dentist to get the cavities treated because less cavities would be missed. The end result could be a disparity in the number of cavities between racial and/or ethnic groups.

References

Cavities . (2023). https://www.cdc.gov/oralhealth/fast-facts/cavities/index.html#:~:text=Among%20adults%20aged%2020%20and,had%20at%20least%20one%20cavity.&text=1%20in%204%20adults%20aged,has%20at%20least%20one%20cavity.

Data Protection and Privacy | AWS. (2023). Amazon Web Services, Inc. https://aws.amazon.com/compliance/data-protection/

Disparities in Oral Health. (2023). https://www.cdc.gov/oralhealth/oral_health_disparities/index.htm#:~:text=Older%20non%2DHispanic%20Black%20or,older%20non%2DHispanic%20White%20adults.

Dutta, S. (2019, September 14). *Why Stochastic Gradient Descent Works? - Towards Data Science*. Medium; Towards Data Science. https://towardsdatascience.com/https-towardsdatascience-com-why-stochastic-gradient-descent-works-9af5b9de09b8#:~:text=Gradient%20Descent%20is%20the%20most,model%20learn%20a%20lot%20faster.

Gandhi, R. (2018, July 9). *R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms*. Medium; Towards Data Science. https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e

How Much Does a Cavity Filling Cost With and Without Insurance? | Humana. (2023). Humana.com. https://www.humana.com/dental-insurance/dental-resources/how-much-does-cavity-filling-cost

Hui, J. (2018, March 27). *Understanding Feature Pyramid Networks for object detection (FPN)*. Medium; Medium. https://jonathan-hui.medium.com/understanding-feature-pyramid-networks-for-object-detection-fpn-45b227b9106c

Nelson, J. (2020, February 5). *When should I enable the grayscale pre-processing setting?* Roboflow Blog; Roboflow Blog. https://blog.roboflow.com/when-to-use-grayscale-as-a-preprocessing-step/

Rashid, U., Javid, A., Abdur Rehman Khan, Liu, C., Ahmed, A., Khalid, O., Saleem, K., Shaista Meraj, Iqbal, U., & Nawaz, R. (2022). *A hybrid mask RCNN-based tool to localize dental cavities from real-time mixed photographic images*. *8*, e888–e888. https://doi.org/10.7717/peerj-cs.888

ResNet-50: The Basics and a Quick Tutorial. (2023, May 22). Datagen. https://datagen.tech/guides/computer-vision/resnet-50/#

Rudy, M. (2023, June 26). *AI helps dentists catch more cavities and gum disease: It's "unbiased" and gives "more accurate" diagnoses*. Fox News; Fox News. https://www.foxnews.com/health/ai-helps-dentists-catch-cavities-gum-disease-unbiased-gives-more-accurate-diagnoses

StepLR — PyTorch 2.0 documentation. (2023). Pytorch.org. https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.StepLR.html

De, S., Zhou, Y., Larizgoitia Abad, I., & Moessner, K. (2017). Cyber–Physical–Social Frameworks for Urban Big Data Systems: A Survey. Applied Sciences, 7(10), 1017. https://doi.org/10.3390/app7101017