Original article

Journal of High School Science

A custom Convolutional Neural Network to streamline oral disease detection and screening

Pericherla A[1], Kalidindi M[2]

Submitted: June 18, 2024, Revised: version 1, July 17, 2024, version 2, July 28, 2024
Accepted: July 29, 2024

Abstract

Oral diseases pose a significant health burden on people across the globe, often causing lifelong affliction and leading to discomfort, pain, and in some cases, death. According to the WHO Global Health Status Report (2022), it is estimated that oral diseases affect almost 3.5 billion people, underscoring the necessity of early identification. Current screening tools are expensive and time-consuming, further arguing for the need to find a favorable alternative. In this research, a Convolutional Neural Network (CNN) was constructed that could accurately detect and classify a variety of oral diseases, including gingivitis, calculus, caries, and cancer. The CNN model made use of a combination of techniques such as transfer learning and dropout layers to provide optimal results. A Kaggle dataset containing 6000 images was used to train, test, and validate the efficiency of the model. This dataset, unlike others, constituted various image orientations and sizes. The model was run both with and without GPU accelerators. The trained and fine-tuned model yielded specificities > 92% for all the six categories of oral diseases. Sensitivities of > 87% were obtained for all categories except for gingivitis, which was detected with a sensitivity of > 54%. Such AI-driven disease prediction and classification approaches have the potential to function as diagnostic aids, increase patient satisfaction, and result in faster, more personalized, and cost-effective oral healthcare solutions. Furthermore, they pave the way for advancements in affiliated pipelines or processes such as disease screening and interactive software use.

Keywords

Oral diseases, Gingivitis, Caries, Machine Learning, Convolutional Neural Network, Transfer learning, GPU acceleration, Dropout layers

---

[1]Corresponding author: Aditya Sai Varma Pericherla, Oakridge International School, Villa 2, Private 28, Kokapet SEZ Road, Kokapet, Hyderabad, Telangana 500075, India. adityapericherla@gmail.com

[2]Madhuri Kalidindi, Kyndryl Solutions, Phoenix Avance Business Hub, SEZ Gachibowli, Hyderabad 500081, India. kalidindim@gmail.com

1 Introduction

Over the past three decades, the estimated prevalence of oral diseases has increased by almost 50%. As of 2020, oral cancer, which is the 13th most common cancer worldwide, has led to more than 177,000 deaths worldwide (1). Most deaths occur in less economically developed countries where there exists inadequate preventative oral healthcare information, education and dissemination; excessive use of tobacco, betel nut, and slaked lime; and grossly underfunded oral healthcare clinics. India is one such country, having the highest prevalence of oral cancer due to the ubiquity of smokers and tobacco chewers (2).

Several prior studies have assessed the accuracy of physician diagnoses of oral cancer. General dentists misdiagnosed 45.9% of oral diseases; oral and maxillofacial experts misdiagnosed 42.8% of diseases; and cancerous lesions were misdiagnosed 5.6% of the time (3). Procedures such as CT scans, X-rays, physical examinations, and biopsies allow a dentist to detect oral diseases, but they are expensive and time-consuming, making it inaccessible to low-income patients and infrastructure/equipment poor facilities. Furthermore, there are inherent limitations on relying solely on human expertise in order to detect oral diseases, as is evident from the data given above.

The primary objective of this project was to construct an accurate and efficient Convolutional Neural Network (CNN) model that could serve as a diagnostic aid in detecting a variety of oral diseases. Computational techniques such as transfer learning, GPU accelerators, custom neural layer architecture, feature engineering and hyperparameter optimization were employed; adapted from deep learning mechanisms, so as to overcome some of the limitations faced by previous research in this field. Another objective of this study was to detect and categorize oral disease (or any other medical disease or disorder that relies on visual diagnosis for that matter) in its early stages, regardless of image angle or camera orientation, and construct Machine Learning screening tools that systematized that process.

2 Materials and Methods

2.1 CNN introduction and Data processing

2.1.1 *CNN introduction*

CNNs consist of multiple layers of artificial neurons that emulate the behaviour of natural neurons and calculate weighted sums of inputs to scores, which are arbitrary predictions as to which class the input image is most likely to belong to (4). Upon feeding an image into this network, each layer extracts features such as corners or edges. Image classification is based on images that are stored in predefined classes. As the layers extract more features of a given image, they gradually start recognizing patterns that align with any of the given classes and associate that image with that class with a certain probability. CNN architecture can be constructed using Transfer Learning wherein weights from a pre-trained model are extracted to identify features from these sets of images (5). Those can then be transposed on hidden layers that perform the task of classifying images previously unseen by the algorithm.

2.1.2 *Data attributes and pre-processing*

The model was trained on 1300 images of calculus, 1000 images of oral cancer, 700 images of caries, 1000 images of gingivitis, 1200 images of hypodontia, and 700 images of noncancer (each dataset contained images populated with a single disease category) (6). An advantage to the dataset used was that it presented with high variance. This was because each category of oral disease consisted of images that were taken from various angles. This particular feature of the data ensured diversity and prevented the model from associating a particular camera angle with a specific category of disease. Each image was converted into a normalized grid of numbers with each number representing a particular color score of that pixel (normalization scales the numbers to a range between -1 and 1). This step is part of a procedure known as pre-processing which starts with cleaning data to ensure that the classes are balanced and that there are no mismatching images and no out-of-context images.

The dataset was fed to the CNN model where a train-validation-test split of 80-10-10 was applied. Once the model finished training on 80% of the images that had been fed in initially, its accuracy was tested on the 10% of validation images. The rest of the images acted as new data that the model had not encountered before. This process ensured that the model possessed the ability to generalize features and then learn from the data, rather than memorize the data; a concept known as 'overfitting.'

2.2 Transfer learning and custom layer architecture

2.2.1 *Transfer learning*

Transfer learning is a machine learning technique used to gain knowledge from a particular task (dataset) and apply that knowledge to a different, yet related task (dataset) (7). To this effect, a ResNet50 model, which is a pre-trained model that consists of 50 layers and is used in image classification problem sets to achieve accurate results, was added to the CNN sequential layer. Part of the reason why the ResNet50 model is as accurate as it is, is because it has been trained on more than 1 million images used in a variety of different image classification tasks. ImageNet is a dataset listing almost 1000 categories of images. Although there are no specific oral cavity or oral disease categories, the process of extracting generic features and attributes from images can directly be transferred from ResNet50 to the oral diseases categories use case. This is because only the foundational feature extraction capabilities are transferred. It also implies that the 1000 image categories that ResNet50 has trained on do not play any role in the Oral diseases CNN model's predictive capacity.

An analogy can be used to understand this process better. (Say) someone has already become proficient at playing the violin. They have learnt how to read music, interpret rhythm, and have developed finger dexterity. Now, if they want to learn the piano, they do not have to start from scratch. Rather, the skills and knowledge that they have gained from playing the violin can be *transferred*. This significantly reduces the time and effort needed to learn the piano as compared to if they were starting from scratch. Transfer learning is somewhat akin to learning to play the piano after becoming a proficient violinist.

The ResNet50 model used was previously trained on Image Net containing 1000 categories of images. It therefore contained 1000 outputs. This model was adopted to the oral disease diagnosis and classification CNN model by freezing its (ResNet50) middle layers so that its weights; or the 'strength' of the nodes; remained unchanged. Its initial layers, on the other hand, were leveraged for their feature extraction capabilities, while the final layers, also known as modifiable or trainable layers, were fine-tuned to perform effectively on the new 6 category oral disease dataset. Custom layers were concatenated to the final ResNet50 layers in the fine-tuning process (explained further in detail) and the dense layer having 1000 outputs (on the original model pre-trained on Image Net containing 1000 categories), was manually replaced by a custom dense layer having 6 possible outputs (for the 6 categories of oral diseases) with one final prediction or class score. Then, this modified architecture was trained on the new dataset of 6000 images. The model adjusted the weights during the process (to minimize the cost function) and became specific to the use case of identification and classification of 6 categories of oral diseases.

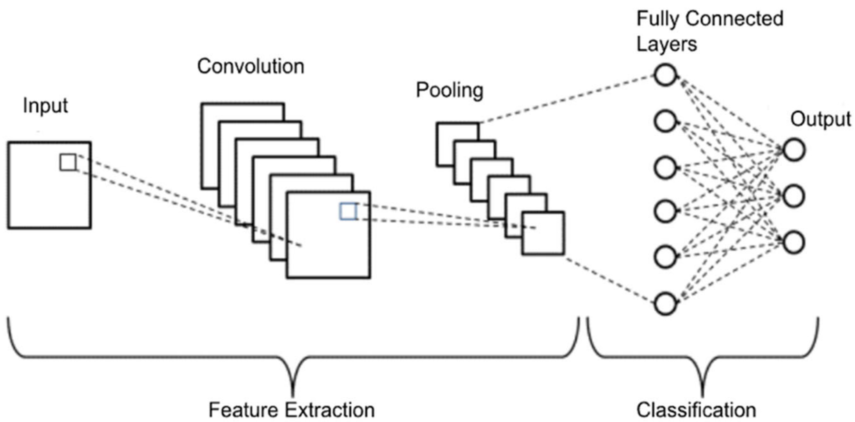

Figure 1. A visualized Convolutional Neural Network, showing the various stages of processing the image and predicting the final output class (8).

2.2.2 *Custom layer architecture*

The custom layers started with a 2D convolutional layer consisting of 32 filters. These filters had a kernel size of (3,3) and a stride of 2 steps. A kernel is a square with a specific feature embedded within it. An advantage of using CNNs is that these features need not be engineered manually but are recognized as the model acquires more training data. For example, one kernel could be looking for a straight line and another could be looking for a horizontal line. The programming interface would then perform matrix multiplication or dot product with each grid of numbers of the image to obtain a set of values on a new grid called a matrix. The ReLU (rectified linear unit) activation function was added to this model to neglect any negative values in the new matrix. This function introduced non-linearity and ensured that not all nodes were activated in the feature extraction process, thereby reducing computational expense and time.

2.2.3 *Pooling and dropout layers*

Subsequently, a pooling layer was used to decrease the size of the final matrix and ensure that the model did not overfit the data. Overfitting is a problem that occurs when a model is overtrained so that it resorts to memorizing features rather than learning from them (9). A max-pooling layer, which was used for this model, selected the maximum value in a matrix of a particular region and created an entirely new reduced matrix with that value representing that region. This reduced the number of values present in the matrix and, in turn, the size of the grid itself. The process therefore increased the flexibility in the model toward variations and distortions in an image owing to a higher rate of generalizability.

This model also incorporated a dropout layer to reduce overfitting (10). This is an important technique that avoids overfitting by dropping out certain features that are extracted from the image to ensure that the model performs effectively even when it is blind to those features. This is also part of a process known as unstructured pruning that creates a sparse network and provides storage and computational advantages.

2.2.4 *Additional Layers*

This process was repeated for 3 more layers with an increasing number of filters; 32, 64, 128 and 256. A progressive increase in the number of filters allowed the model to extract an increasing number of features, increasing its identification and classification accuracy. A flattening layer was used to convert the entire matrix into a single array and act as a bridge between the feature extraction and classification layers.

The classification layers consisted of dense layers which are densely connected neurons that relay the scores of an image with each of the corresponding classes, or in other words, its probability of being associated with one class over another. A SoftMax activation function was used in this dense layer. This activation function is an extension of logistic regression, a process by which the probabilities of a specific image being associated with a particular class are output. SoftMax simply extends that functionality to multiple classes (in this case, 6 classes) and ensures that these probabilities add up to 1.0 (11).

2.2.5 *Hyperparameters selection*

Before training the model on a certain number of iterations, an optimizer, a loss function, and a reduction in learning rate were chosen. The Adam optimizer, which is short for “Adaptive Moment Estimation”, is used in conventional practice to reduce the loss of class scores (12). It was chosen after running the data through preliminary iterations with other optimizers such as RMSprop (13).

A learning rate reduction via the ‘Reduce Learning rate on Plateau' callback mechanism was initiated to monitor the model's validation accuracy of the model. It output information concerning the metric's performance and improved the model’s convergence, i.e., its ability to interpret data reasonably (14). Sparse_categorical_crossentropy is a loss function that measures the probability between the predicted and true probability distributions. This method was chosen because it was compatible with multi-class classification problems (15).

2.2.6 *GPU accelerators*

The model was run with GPU accelerators (GPU T4 x2). These are hardware computing resources that accelerate or speed up the analysis of data and of certain operations such as matrix multiplication. The accelerators allowed the model to train and test the data quickly and efficiently (16).

Figure 2. Sample images of oral cancer, calculus, caries, gingivitis, no tumor, and hypodontia (from left to right starting from the top row). Each category of disease consisted of images having different camera angles; a sample from each is shown.

3.1 Training and validation speeds

3.1.1 *Speed enhancement with GPU accelerator*

As shown in Table 1, the model performed 24 times quicker with a GPU accelerator while training and validating the model and 95 times faster with a GPU accelerator to test the image.

Table 1. Training and validation and testing speed (with and without GPU accelerator)

Speed of:	With GPU accelerator	Without GPU accelerator
Training and Validation	8 minutes 20 seconds	3 hours 20 minutes
Testing a single image	0.4 seconds	38 seconds

3.2 Accuracy metrics

3.2.1 *Accuracy*

After running for 25 epochs, the model reached an overall accuracy was 95%, which was the highest on the 10th epoch. The choice of an epoch varies from study to study and is a hyperparameter that must be tested to ensure that the model can learn underlying patterns without overfitting the data. The training accuracy peaked at 95% on the 19th epoch. The greatest validation accuracy was 85% on the 20th epoch.

3.2.2 *Confusion matrix, ROC, and loss*

The confusion matrix showing the number of images classified correctly and incorrectly for each of the classes is shown below.

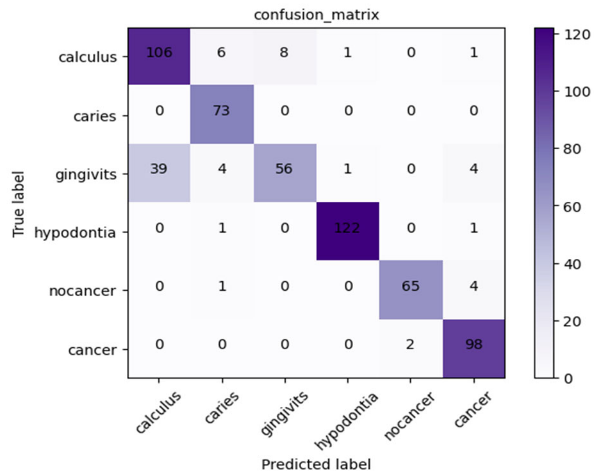

Figure 3. A confusion matrix for the classes (image generated on Kaggle).

The individual validation accuracies of each of the classes were plotted using the confusion matrix. Out of 100 images in the cancer test set (by adding the numbers in each row for each class), 98 were predicted to be true thereby resulting in an accuracy of 98%. The lowest accuracy of 53.8% was obtained for gingivitis. The general formula for this qualitative analysis is given below.

$$Accuracy = \frac{n(Truelabel \cap Predictedlabelofaclass)}{Totalnumberofimages \in class} = \frac{98}{100} \qquad (1)$$

This method applied to each of the other classes resulted in an accuracy of 92.9% for the category of no cancer (free of other diseases as well), 98.4% for hypodontia, 53.8% for gingivitis, 100% for caries, and 86.9% for calculus.

In addition, the AUC score was 0.98. The ROC curve shows the true positive rate of the model mapped against the false positive rate. This calculation was performed by using an AUC Python package where successive trapezoids and polygons are added to obtain the area under each category curve.

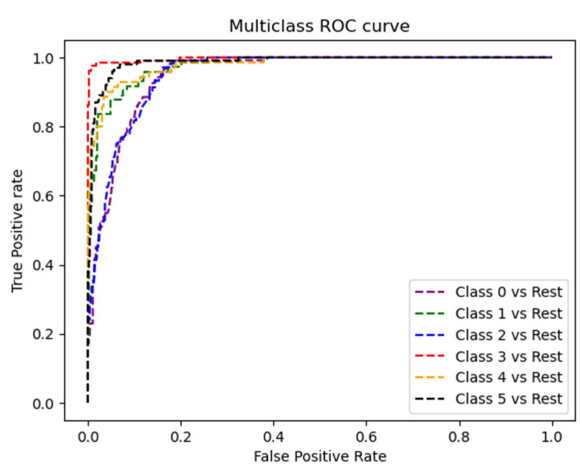

Figure 4. ROC curve showing the AUC of each of the classes 0, 1, 2, 3, 4 and 5 which correspond to categories ‘calculus', 'caries', 'gingivitis', 'hypodontia', 'no cancer', and 'cancer', respectively (image generated on Kaggle).

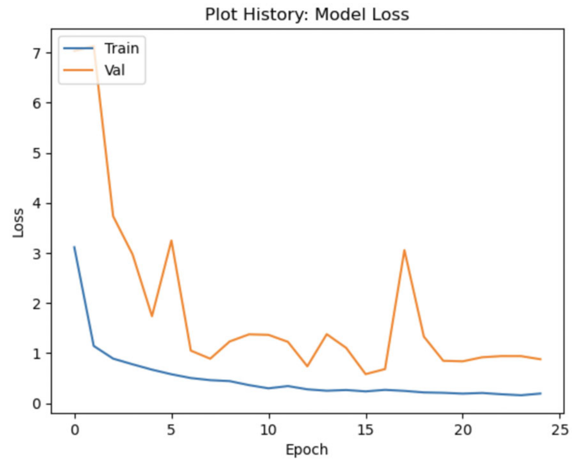

Figure 5. A plot of the change in loss over epochs for the train and validation datasets (image generated on Kaggle).

A loss function allows an estimation of the model's accuracy. A greater loss function implies a lesser accuracy. For the model used in this study, the loss decreased for both the training and validation data and reached a value of ~ 0.1 for the training dataset, and ~ 1 for the test dataset. The 5th and 17th epochs saw spikes in the loss function which were inconsequential to the final accuracy of the model.

3.2.3 *Sensitivity and Specificity*

Table 2 shows the sensitivity and the specificity of the model for identification and classification of various oral diseases. Sensitivity defines the ability of a model to correctly identify patients with a disease. Specificity defines the model's ability to correctly identify people without the disease. The sensitivity and specificity of the model can be calculated using the confusion matrix.

Table 2. Sensitivities and specificities of the model along with threshold attributes

Disease	Sensitivity	Specificity	Threshold attributes
Calculus	0.87	0.92	Yellow brown deposits, white plaque
Caries	1	0.98	Discoloration, dark spots
Gingivitis	0.54	0.98	Inflammation of gums, bleeding gums (red saturation), swelling, plaque
Hypodontia	0.98	0.99	Missing teeth, gaps in enamel structure and tooth placement
No cancer	0.93	0.99	Consistent coloration, absence of abnormal growth
Cancer (oral)	0.98	0.98	Discoloration, abnormal growth (lesions)

$$\text{Sensitivity: } \frac{TP}{TP+FN} \tag{2}$$

$$\text{Specificity: } \frac{TN}{TN+FP} \tag{3}$$

The true positive rate (TP) is the probability that an image presenting with disease in a particular class was predicted as being positive for the disease. For oral cancer, this is 98, as can be seen from the confusion matrix. The false negative rate (FN) is the probability that an image with a true label was missed, in the case of oral cancer this is 2. The true negative rate

(TN) is the probability that an image that shows no disease for that class will be correctly predicted as one that does not present with disease. For oral cancer, this is 483 (one can arrive at this by adding all the numbers that are not part of the row and column, in the confusion matrix, associated with the class at hand). The false positive rate (FP) is the probability that an image that presents with no disease is falsely classified as one that is diseased. This happens 10 times for oral cancer. This process was repeated for each class to obtain the results in Table 2.

4 Discussion

4.1 General Discussion

The CNN model used in this study obtained individual testing class accuracies of 92.86% for no cancer (and free of the other diseases), 98.39% for hypodontia, 53.85% for gingivitis, 100% for caries, and 86.89% for calculus.

Some general observations follow from these results. Firstly, the accuracy of the model was higher than the average accuracy for doctors to detect these diseases (3). The model however, was limited by the amount of data that it could access. Since a public dataset was used there were only 6000 images for all the classes.

Gingivitis presented with a low accuracy. Most of the images in this category were misclassified as calculus. A reason most attributable to this anomaly could be that the colour of some of the images had been altered: they were of high contrast and tinted red, emulating rather extreme lighting and/or colour conditions. Another possibility is that the image presented with multiple diseases and hence led to a misclassification, however, this reason can be disregarded as each image presented with only a single disease. Indeed, the use of this gingivitis disease dataset with the red tinted images was debated to be included originally, but, ultimately, it was tested to showcase the significance of image quality and colour variations in accurate disease classification. In practice, such data would need to be pre-processed or, in extreme situations, avoided altogether. Imaging software could have been used to remove the tint in the images of gingivitis rather than deleting the images altogether from the dataset. However, due to the sometimes poor quality of images in this open source dataset, this manuveur could potentially have degraded the image further. The two images shown in Figure 6 both present with gingivitis but the one in the right panel was incorrectly classified due to the image being tinted red.

Figure 6. Two images both containing gingivitis and taken from the same angle. The image on the right was misclassified due to a red tint. This decreased the model's predictive capacity (most of the misclassified images presented with the red tint).

To test this hypothesis, 200 images (out of 1000) that were tinted red or presented with a high colour contrast were removed from the training and the test datasets. These were run through the CNN model again. This time, the model classified the images with gingivitis as correctly being diseased with gingivitis, with an accuracy of 82.6%, significantly greater than the accuracy of 53.8% that was obtained when all the 1000 images were run through the CNN. Therefore, the effect of image colour on image classification tasks appears to be non-trivial and needs investigation in future ML models.

Figure 7. Part of the Confusion matrix showing the gingivitis class after removing the tinted images (Kaggle generated image).

A plausible reason as to why gingivitis had been predominantly misclassified as calculus could be due to the fact that image symptoms for these diseases may overlap. Calculus, which presents in images as hardened plaque on teeth, if left untreated and unremoved usually leads to gingivitis, among other complications. Since the model is unable to classify a single image into multiple categories, a misclassification occurs. This may also be why some calculus images were misclassified as being presented with gingivitis.

Another limitation of this model is its inability to accurately classify images of teeth that contained tobacco residuals or coloured fluid/food deposits. This may have occurred for the same reason that gingivitis images with colour variations were misclassified; i.e. they acted as outliers which decreased the model's ability to associate the image with any particular pattern.

4.2 Comparison of the model with ML used in the published literature

In a study by Almaki, a YOLOv3 model achieved an accuracy of 99.33% for dental disorders such as cavities and root canals, among others, even with the use of fewer images (1200) (17). A feature of object detection that proves to be quite advantageous, is its ability to locate the regions in which the model predicts that the hallmarks of disease may be present. This can allow the model to perform multi-class classification wherein multiple diseases can be detected within the same image. The CNN model constructed in this study could not detect multiple diseases in one image. Object detection could conceivably be incorporated into the model used in this study. The anticipated outcome would be that during the training phase, this feature could be harnessed to flag incorrect classifications, locate the region which may be an indicator of a particular disease, and provide invaluable insight to a clinician or doctor who may wish to understand the functioning of the model. Regardless, incorporating the object detection method into the CNN model used in this study would need to be further validated because of the paucity of images that, in turn, could lead to overfitting. To some extent, the problem of overfitting because of using a small dataset was overcome in this study by incorporating dropout techniques; optimization software; and pooling layers. The ability of the model to accurately classify oral disease with a > 87% sensitivity (with the exception of gingivitis) demonstrates its competitiveness in the landscape of ML oral disease detection models. Furthermore, it overcomes a limitation that is prevalent in the published literature, namely, the lack of a diverse dataset defined by images captured using variable camera/imaging angles, orientations and/or configurations. This induces generalizability and cross-applicability, features missing in the published literature.

Huang et. al. implemented a seagull optimization algorithm to provide an accuracy of 96.94%. This accuracy was greater than models using FCM, CNN, R-CNN, and ResNet-101 (18). This unique method efficiently optimized the structural hyperparameters of a CNN architecture. We optimized these hyperparameters as well but used other techniques such as Sparse_categorical_ crossentropy (https://github.com/keras-team/keras/blob/v3.3.3/keras/src/losses/losses.py#L1677-L1738) and ReduceLROnPlateau (https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.ReduceLROnPlateau.html#reducelronplateau). Huang et al. also implemented advanced image enhancement techniques, which differentiated it from our study. Data augmentation is another common technique that is used in several such neural networks that allows the architecture to adapt to changing light conditions, image sizes, image rotations, image blur, among other such variations. Though it proves to be advantageous this technique was not incorporated in this study for two main reasons. Firstly, since the data used was already quite diverse and sufficient, the use of data augmentation was expected to provide little to no advantage in terms of the accuracy of the model. Secondly, if data augmentation is not meticulously monitored, it may introduce unexpected image noise and diminish the quality of the dataset which will, inadvertently, lower the model's performance.

4.3 Perspective

To expand the scope of this project and possibly commercialize its use, a screening tool can be integrated into the process of oral disease detection that implements such a model. Using this screening tool may dispense with the need for initial scans. This is because taking photographs of the teeth, or the entire mouth, either at home or at the clinic will serve as well. These pictures can then directly be run through the CNN algorithm screening tool.

In considering that possibility, the need for a graphical user interface (GUI) that creates an effective channel of communication between a patient and a doctor, is indispensable.. The process of designing such an interface must be conceptualised keeping intuition and familiarity in mind. The image shown below is an example of such an interface taken from a python library: Gradio (https://www.gradio.app/). This interface requires attaching an image and clicking Submit; the Convolutional Neural Network runs in the background and generates a predictive class output. The interpret button allows the developer to assess which part of the image the model is interpreting and possibly 'flag' that instance. In a clinical setting this web application could be used to communicate and explain the working of such a model with the patient as well. This is currently an overlooked part of a machine learning driven diagnostic process.

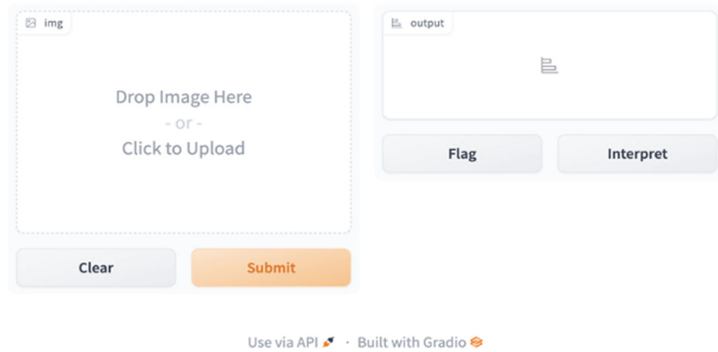

Figure 8. Gradio interface that allows attaching an image and generating the predicted output class (in this case, the disease name) with a specific probability (19).

A multi-faceted strategy must be employed to overcome all the challenges presented above. For instance, the implementation of an object detection model that can detect and classify multiple diseases within the same image shows promise in the future of machine learning based diagnostic approaches. Subsequently, supervised or semi-supervised learning may also provide a robust strategy in enhancing the model's performance by reducing its dependence on labeled data and taking advantage of unlabeled data as well. The fact that the model cannot perform effectively on a novel disease warrants the need for a more exhaustive and thorough dataset that comprises of a number of different oral diseases. In response to that challenge, the possibility exists of creating blocks of the same model. These blocks could then be fine-tuned for a specific disease; a focused attempt at improving the model's performance for each disease.

5 Conclusion

In this study, custom layers were added to a pre-trained ResNet50 model to classify oral diseases into six categories: gingivitis, oral cancer, no-cancer, hypodontia, caries, and calculus. The individual class sensitivities were 92.86% for no cancer (and free of the other diseases), 98.39% for hypodontia, 53.85% for gingivitis, 100% for caries, and 86.89% for calculus. The algorithm is imaging/camera angle, orientation and/or configuration agnostic. Hence, this Convolutional Neural Network-based approach for identifying and classifying oral diseases has no requirement for expensive or inaccessible imaging techniques or equipment; instead working off of patient-generated photographs or images. It is hence amenable for use as part of an intuitive screening tool to provide oral care to underserved and impoverished populations.

6 References

1. Tranby E. P., Heaton L. J., Tomar S. L., Kelly A. L., Fager G. L., Backley M., et al. Oral cancer prevalence, mortality, and costs in Medicaid and commercial insurance claims data. Cancer Epidemiol Biomarkers Prev. 2022, 31(9):1849–57. http://dx.doi.org/10.1158/1055-9965.epi-22-0114

2. Oral Cancer. Indiancancersociety.org. https://www.indiancancersociety.org/oral-cancer/

3. Kondori I, Mottin RW, Laskin DM. Accuracy of dentists in the clinical diagnosis of oral lesions. Quintessence Int. 2011, 42(7):575–7. https://pubmed.ncbi.nlm.nih.gov/21716985/

4. Oral health conditions. Cdc.gov. 2023 https://www.cdc.gov/oralhealth/conditions/index.html

5. Das P., Sony D., Thimmaraju M. K., Ahalya N., Anandaram H., Kumar A. Design and comparison of transfer learning for dental caries detection. In: 2022 International Conference on Edge Computing and Applications (ICECAA). IEEE; 2022. p. 1311–6. https://doi.org/10.1109/ICECAA55415.2022.9936165

6. Sajid S. Oral Diseases. 2024. https://www.kaggle.com/datasets/salmansajid05/oral-diseases

7. What is transfer learning? GeeksforGeeks. 2019. https://www.geeksforgeeks.org/ml-introduction-to-transfer-learning/

8. Gurucharan M. K., Basic CNN architecture: Explaining 5 layers of Convolutional Neural Network. upGrad blog, upGrad Education, 2022 https://www.upgrad.com/blog/basic-cnn-architecture/

9. Amazon.com https://aws.amazon.com/what-is/overfitting/#:~:text=Overfitting%20occurs%20when%20the%20model,to%20several%20reasons%2C%20such%20as%3A&text=The%20training%20data%20size%20is,all%20possible%20input%20data%20values

10. Dropout layer. Keras.io. https://keras.io/api/layers/regularization_layers/dropout/

11. Multi-class neural networks: Softmax. Google for Developers. https://developers.google.com/machine-learning/crash-course/multi-class-neural-networks/softmax

12. Optimizers. Keras.io. https://keras.io/api/optimizers/

13. Verma P., Tripathi V., Pant B. Comparison of different optimizers implemented on the deep learning architectures for COVID-19 classification. Mater Today. 2021, 46:11098–102. http://dx.doi.org/10.1016/j.matpr.2021.02.244

14. ReduceLROnPlateau. Keras.io. https://keras.io/api/callbacks/reduce_lr_on_plateau/

15. Tf.Keras.Losses.Sparse_categorical_crossentropy. TensorFlow. https://www.tensorflow.org/api_docs/python/tf/keras/losses/sparse_categorical_crossentropy

16. Cornell.edu. https://www.cs.cornell.edu/courses/cs4787/2019sp/notes/lecture25.pdf

17. Almalki Y. E., Din A. I., Ramzan M., Irfan M., Aamir K. M., Almalki A., et al. Deep learning models for classification of dental diseases using orthopantomography X-ray OPG images. Sensors (Basel). 2022, 22(19):7370. http://dx.doi.org/10.3390/s22197370

18. Huang Q., Ding H., Razmjooy N. Oral cancer detection using convolutional neural network optimized by combined seagull optimization algorithm. Biomed Signal Process Control. 2024, 87(105546):105546. http://dx.doi.org/10.1016/j.bspc.2023.105546

19. Image classification in tensorflow. Gradio.app. https://www.gradio.app/guides/image-classification-in-tensorflow