Oral Disease Classification Model Report

By Nandan Wahi (nandanwahi@gmail.com)

1. Introduction

- **Project Overview**: The objective of this project is to develop a deep learning model to classify oral diseases, such as dental caries, gingivitis, and healthy oral conditions, from images. The model uses Convolutional Neural Networks (CNN) to process images and classify them into the appropriate categories, providing a useful tool for diagnostic support in the medical field.
- **Objectives**:
  - The model performs **multiclass classification** to classify each image into one of three categories: Caries, Gingivitis, or Healthy.
  - The model should handle variations in image quality, lighting, and orientations to ensure robust classification across various demographic groups.
- **Tech Stack**:
  - **Python**: Primary programming language used for model development.
  - **TensorFlow/Keras**: Deep learning framework used for building and training the CNN.
  - **OpenCV**: For image processing (if needed for additional preprocessing).
  - **Matplotlib/Seaborn**: Used for visualization (loss/accuracy curves and confusion matrix).

2. Dataset Description

- **Dataset Overview**: The provided dataset contains only two categories: **Caries** and **Gingivitis**. It was divided into training and test sets for these two categories. However, the **Healthy** category was missing from the dataset. To overcome this limitation, a **web scraper** was developed to gather additional images for the "Healthy" class from **Google Images**.
  - **Web Scraping for Healthy Teeth Images**: Using a Python-based web scraper, several images of healthy teeth were collected from Google Images to supplement the dataset.
  - **Additional Data from Kaggle**: To further enrich the "Healthy" class, images were sourced from the **Dental Anatomy Dataset** available on Kaggle (https://www.kaggle.com/datasets/saisiddartha69/dental-anatomy-dataset-yolov8). The dataset provided anatomical images of healthy teeth, which were handpicked to ensure quality.
  - **Image Processing and Selection**: The images collected for the "Healthy" class were handpicked for relevance and quality. As a result, many of the images in the "Healthy" images folder, found in both the training and test datasets, were **duplicates**. This duplication was intentional to provide a sufficient number of images for the model.

  - **Number of classes**: 3 (Caries, Gingivitis, Healthy)
  - **Image Dimensions**: Each image was resized to 150x150 pixels to standardize the input to the model.
- **Data Preprocessing**: The preprocessing steps include:
  - **Rescaling**: Images are rescaled to a range between 0 and 1 by dividing by 255.
  - **Augmentation**: The training data is augmented using random transformations such as:
    - Rotation range (up to 20 degrees)
    - Width and height shifts (up to 20%)
    - Shear, zoom, and horizontal flips
  - **Target Size**: All images are resized to 150x150 pixels to maintain consistency across the dataset.

3. Model Development

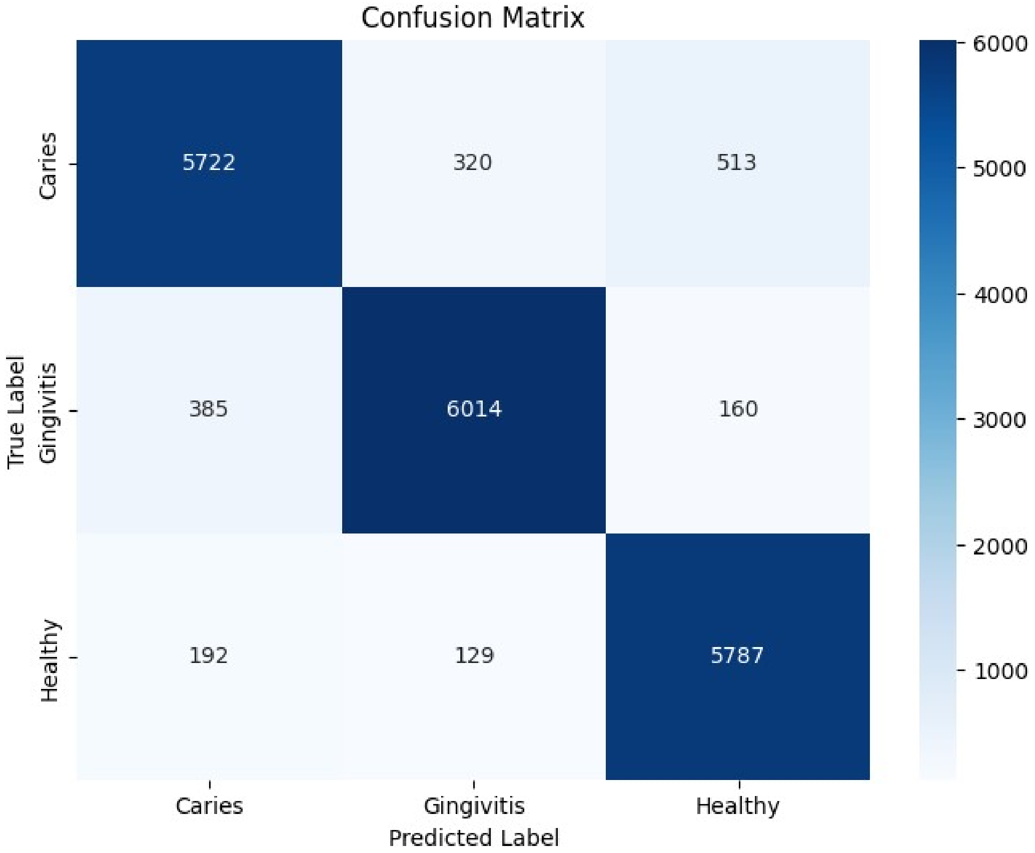
- **Model Architecture**: The model is a Convolutional Neural Network (CNN) built using Keras. The architecture is as follows:
  1. **Conv2D Layer**: 32 filters, kernel size (3,3), ReLU activation.
  2. **MaxPooling2D Layer**: Pooling with size (2,2).
  3. **Conv2D Layer**: 64 filters, kernel size (3,3), ReLU activation.
  4. **MaxPooling2D Layer**: Pooling with size (2,2).
  5. **Conv2D Layer**: 128 filters, kernel size (3,3), ReLU activation.
  6. **MaxPooling2D Layer**: Pooling with size (2,2).
  7. **Flatten Layer**: Flattens the output from the convolutional layers into a 1D array.
  8. **Dense Layer**: 512 units, ReLU activation, followed by a **Dropout Layer** (50% dropout rate) to prevent overfitting.
  9. **Final Dense Layer**: 3 units (for 3 classes), Softmax activation to output probabilities for each class.
- **Model Compilation**: The model is compiled using:
  - **Adam Optimizer** with a learning rate of 0.001
  - **Categorical Cross-Entropy Loss** function for multiclass classification
  - **Accuracy** as the evaluation metric.

Training Process:

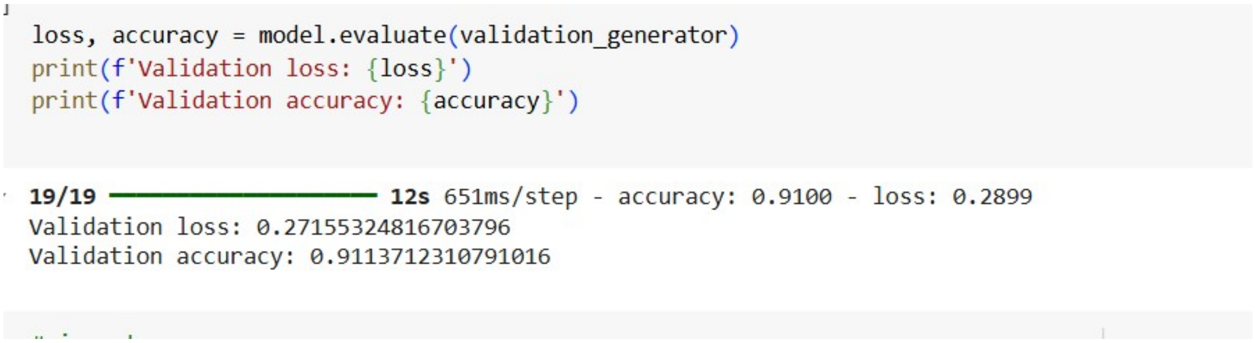
  - **Epochs**: 15
  - **Batch Size**: 32
  - **Callbacks**: Early stopping is applied with a patience of 5 epochs to avoid overfitting. The best model is saved using ModelCheckpoint based on validation loss.

4. Model Evaluation

- **Evaluation Metrics**: The model's performance was evaluated using the following metrics:
  - **Accuracy**: 0.9116116949328894
  - **Precision:** 0.9119525433040635
  - **Recall:** 0.9116116949328894
  - **F1-Score:** 0.9114147891727106
  - **Confusion Matrix**:



- **Results**:
  - **Validation Loss**: 0.27155
  - **Validation Accuracy**: 91.137%
  - **Classification Report** (Example results):



```
loss, accuracy = model.evaluate(validation_generator)
print(f'Validation loss: {loss}')
print(f'Validation accuracy: {accuracy}')
```

```
19/19 ━━━━━━━━━━━━━━━━━━━━ 12s 651ms/step - accuracy: 0.9100 - loss: 0.2899
Validation loss: 0.27155324816703796
Validation accuracy: 0.9113712310791016
```

5. Model Deployment

- **Deployment Process**: The model was deployed in a simple web interface using Flask, where the user can upload an image, and the model will classify the image into one of the three disease categories.
- **Deployment Video**: A video demonstrating the model's deployment and its performance on test images is provided. It shows how the model makes real-time predictions on new images uploaded by users.

6. Conclusion

- **Summary of Findings**: The CNN model achieved an accuracy of **91.137%** on the validation dataset, demonstrating solid performance in classifying oral diseases. The confusion matrix and classification report reveal that the model performs well across the three classes, with the highest accuracy in classifying healthy images.
- **Future Work/Improvements**:
  - **Improvement in Dataset**: The model can be further improved by using a larger and more diverse dataset, including images from different demographics and lighting conditions.
  - **Transfer Learning**: Using pre-trained models like VGG16 or ResNet could improve performance by leveraging learned features from larger datasets.
  - **Hyperparameter Tuning**: Further tuning of hyperparameters such as the learning rate, dropout rate, and the number of epochs could lead to better model performance.

7. References

- Kaggle Dataset: Dental Anatomy Dataset on Kaggle
- Google Images scraping method: Custom scraper (Python-based)

8. GitHub Repositories

The complete project code and model hosting can be found in the following GitHub repositories:

1. **Python Code (Model Training & Evaluation)**:

   https://github.com/Nandan-W/Dental-Disease-Classification

2. **Frontend Code**:

   https://github.com/Nandan-W/Dental-Disease-Classification-Frontend

3. **Backend Code**:

   https://github.com/Nandan-W/Dental-Disease-Classification-Backend