Department of Electrical and Computer Engineering
North South University

Senior Design Project

Oral Cancer Prediction from Images using Convolutional Neural Network

Md. Imtiaz Hoque **ID # 1811330042**

Md. Jahirul Islam **ID # 1821368642**

Md. Azmi Siddique **ID # 1812307042**

Faculty Advisor

Dr. Mohammad Monirujjaman Khan

Associate Professor

ECE Department

Summer, 2022

DECLARATION

This is to certify that this Project is our original work. No part of this work has been submitted elsewhere partially or fully for the award of any other degree or diploma. Any material reproduced in this project has been properly acknowledged.

Students' name & Signature

1. **Md. Imtiaz Hoque**

_____

2. **Md. Jahirul Islam**

_____

3. **Md. Azmi Siddique**

_____

APPROVAL

The capstone project entitled "**Detection of Oral Cancer Using Deep Learning Methods**" by **Md. Imtiaz Hoque (ID#1811330042), Md. Jahirul Islam (ID#1821368642),** and **Md. Azmi Siddique (ID #1812307042)** is approved in partial fulfillment of the requirement of the Degree of Bachelor of Science in Computer Science and Engineering on February and has been accepted as satisfactory.

Supervisor's Signature

Dr. Mohammad Monirujjaman Khan
Associate Professor
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh.

Department Chair's Signature

Dr. Mohammad Rezaul Bari
Associate Professor & Chair
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh.

ACKNOWLEDGMENT

First of all, we would want to thank the Almighty for providing us the stamina to carry out our duties and finish the report.

As a part of the Bachelor of Science (BSc) curriculum, the capstone project program is particularly beneficial in bridging the gap between theoretical knowledge and real-world experience. The purpose of this report is to provide theoretical knowledge through practical experience.

We also want to express our sincere gratitude to all of the teachers who helped us by giving us the technical know-how and spiritual support we needed to finish the project thoroughly.

We must express our gratitude to Dr. Mohammad Monirujjaman Khan, an honorary member of our department, for his undivided attention and assistance in achieving this milestone. Additionally, we are eternally grateful to the ECE department at North South University for offering us with a course like CSE 499 that allowed us to work on this project in depth and bring it to life the way we had imagined.

We are grateful to our friends and family for helping us with this effort and for always being there for us.

ABSTRACT

Oral malignant development is an unavoidable and difficult illness with severe consequences. In India, oral cancer is the seventh most common illness, with 130,000 deaths per year. Growth occurs in the salivary organs, tonsils, neck, face, and mouth. There are several symptomatic tactics for oral disease; for example, a biopsy, in which a little tissue sample is removed from a component of the body and examined under a magnifying lens, as well as certain screening procedures. However, the disadvantage is that it can't clearly detect malignant growth cells and can't describe the number of cells affected by illness, so in this work, malignant growth cells will be discovered and arranged that are impacted in the oral area by computerized handling innovation. For early location and order, cutting-edge innovations and deep learning calculations are viable options. In recent years, cutting-edge technologies like machine learning, deep learning, and convolutional neural networks (CNN) have been employed to solve a number of medical image-related difficulties, including classification. According to CNN, this study employed histological images to differentiate between two types of oral cancer: oral squamous cell carcinoma (OSCC) and oral benign tissue. The purpose of this study is to use a deep, convoluted neural network custom model to classify OSCC and oral benign. The network is trained and evaluated using T1-weighted contrast-enhanced pictures. The network's performance was evaluated using 10-fold cross-validation, and the network's improvement was quantified using augmented pictures. The suggested network exceeds the competition, with an average accuracy of 99.33 percent. The model has a 98 percent accuracy without cross-validation. Furthermore, the record-wise cross-validation approach outperformed the 10-fold cross-validation method, with an average accuracy of 99.33 percent.

Table of Content

List of Figures

List of Tables

CHAPTER 1: INTRODUCTION

1.1 Introduction

Oral illness has been linked to an increase in the number of cells that give off affected surrounding tissues [1]. Oral illness identifies fewer dormant cells in the oral tissue before the progression of oral disease known as ulcers. When digestion occurs, dead cells become accessible in a remote part of the blastic area or inside the body. There are several types of illness, with 90% of crab cells being classified as OSCC (Oral Squamous Cell Carcinomas) [2]. Organic models, as well as clinical forms of associated and painful free growth models, have the potential to be identified in various areas of the body via appearance models and generalizations without staining. AI techniques were used to estimate various organic models for OSCC, which would rank non-carcinogenic and harmful instances, which were then evaluated for the oral malignant development stage [3]. The indicator will determine the precision of the partnership by employing three avocation test units as well as distinct illness stages. A variety of tests can be used to determine specific cancer volumes as well as the presence of ulcers in the tissues, assisting in the prediction of various stages of oral illness [4]. The primary goal of the continuing method is to create new tools for predicting the stage of development of oral malignant growth tumors. Oral cancer develops in areas such as the front of the tongue, the upper and lower sections of the mouth, within the cheeks alongside the lips, the gums, and the area at the back of the wisdom teeth. The following are the negative effects of oral malignant growth: the most common signs of sickness are aggravation or ulcers that don't heal, which might cause anguish in any case of draining [5]. Risk factors for oral cancer include a variety of activities, such as smoking and drinking. These two propensities are thought to be the most significant risk factors for oral illness. Eating worms is so common in India that it affects the gums [6]. The final gambling prospects for Human Papillomavirus (HPV), for example, as well as gender and age. The specialists' goal was to estimate the early stages of oral malignant development using less correct

results from Nave Bayes, Multilayer Perceptron, KNearest Neighbors, and Support Vector Machine algorithms. Oral examination and examination improve order precision. [7]. Ahmad, LG et al. The goal of the scientist is to create a model for doctors. Using tree-based dynamic approaches, forgery brain network vector support procedures, and high-precision examination of DATA, NN, and HDM. The ADMS rating depiction is usually fantastic for finding bosom malignant growth return, with the most increased exactness and the least blunder degree. When compared to ANN and DT, the results suggest that SVM is the most commonly used approach for determining [8]. The goal of Harikumar Rajaguru and Sunil Kumar Prabhakar [9] is to evaluate the location exactness of TNM stage frameworks using Multi-Layer Perceptron (MLP) and Gaussian Mi Mixed Models. The evaluation of the two placement gatherings here produced better findings than the stage's typical exactness. Bright filtering machines (LMMs) are used as post-information capacity gadgets for the examination of oral malignant development, and the viability of LMS classifiers is compared to the efficacy of SMMs and MLPs. Amy F. Z. B. et al. By evaluating patient articulation and RNA extraction, as well as micro-examination, the SVM characterization approach was used to search for OSCC malignancies [10]. The goal is to evaluate innovative computerization techniques for OSCC diagnoses that involve top-to-bottom preparation and CNN strategies on clear imaging. CNN's technique here is to search for remarks, images, preparations, information, and categorization [11]. For example, top-to-bottom ANA analysis, move review, and CNN were employed. In this manner, they attained an accuracy of 88.46% [12]. For Martin Halicek and others to detect illness, analysts ran the OSCC model via CNN. C. N. Xi achieved 80% accuracy in detecting illness [13]. The analysts' goal was to order dental X-beams of damaged and normal teeth, as well as to create imaginative models for X-beam troubles employing advanced neurological innovation [14]. Konstantina Kuru and colleagues compared the AI systems to several classes of malignant growth predictors. The learning used inherent data to demonstrate using AI methodologies as well as

decision tree techniques for the selection and arrangement of attributes [15]. The goal of specialists Wafaa K. Jokes and Zaw Z. Htike was to estimate the movement of oral illness in OPL patients using AI methods applying hereditary portrayal. The specialists used SVM, MLP, minimally invasive daily schedule (RLS), and profound brain organization to investigate the migration of oral illness in patients with OPL records [16]. [17] discusses the significance of the illness disclosure process. They presented the distinct approaches to disclosure technique on various illness points as well as illustrated the method of counterfeit treatments aiding with further growing malignant growth discovery. They carried out the order of brain malignancies by applying normalized test images of UCI's focus on informative indexes. As a result, they have identified health concerns in the financial populace. They were 87.2% accurate in identifying cerebral illness by deep brain organizations. Phillips and colleagues describe an exploration of the many actions of preparation, both within and outside, for the employment of various displaying tactics used to organise clinical images. Furthermore, PC-based PC approaches that enable characterization are being investigated. The notion by Duke Kumama et al. [18] compares the use of speculative imaging for malignant growth disclosure to traditional methods. They evaluated a few strategies for illness detection as well as defined the advantages of suggestive reenactments. As a result, HSI may be used by arranging specialists. They carry out the order using a vector support machine and a self-mapping structure. IT was created by [19] and Dul Al. [20]. P. IT. MRI and CT are other imaging methods such as MRI and CT. They also use calculations to recognize cerebrovascular disease for top-to-bottom preparedness in their training. Wang, L. investigates semi-computerized solutions for illness classification based on top-to-bottom learning calculations. [21]. They used organized power machines for order in the work of Kalantari et al. [22] to improve the use of hyperspectral imaging to diagnose lunge illness. They used CNN to divide it into images. The absence of a totally independent strategy of the malignant growth recognition framework through top-to-bottom

evaluation processes has been brought out on this basis. Most processes want advanced framework setup, which results in significant framework work charges. This study overcomes all of the obstacles identified by Yuan et al. [22] by applying a relapse development of the brain organization of electronic malignant growth disclosure in current clinical imaging with this novel top-to-bottom learning approach. According to the related study, the most recently used techniques and materials are mostly covered for the detection of malignant development, illness categorization, and the evaluation of AI approaches. As a result, oral disease position is an important goal in identifying oral malignant development. This task is not done by a researcher, which is the most significant mission in the examination of expectations alongside the restoration of malignant growth sufferers for specialists. As a result, the current study focuses on the use of a variety of AI approaches that emphasize powerful stage evaluation in the progression of oral malignant development.

Using various approaches, the majority of the examinations evaluated have the potential to attain over 90% precision. Regardless, order accuracy and the normal time required to achieve high precision remain a hurdle. Particularly with regard to machine learning computations, which make use of extremely thick organizations. They are prepared to prepare north of 100 million borders [15]. Furthermore, in order to function properly, these models require a significant amount of processing power, such as high-quality GPUs and a large amount of RAM. In a place like Bangladesh, where there aren't many PCs, this strategy might make establishing a long-term oral illness characterization framework difficult. The current analysis suggests a spectacular CNN engine to improve the accuracy of oral illness categorization. The new CNN model will seek to eliminate manual effort in clinical practice. Furthermore, the proposed technique makes use of fewer registration assets to correctly order oral disease kinds. The remaining sections of this work are as

follows: Section 2 discusses the technique and materials. Section 4 discusses the conclusion, whereas Section 3 delves into the outcome.

CHAPTER 2: MOTIVATION

2.1 Introduction

This chapter discusses the reasoning for our decision to build this system. In this chapter, we will also describe why we picked the medical area over all others to work in.

2.2 Motivation towards our project

Oral cancer is a big problem these days, but because it is not visible in the early stages, most people ignore it until it is too late. And the cost is prohibitively expensive. We can treat it swiftly and cheaply if we catch it early. As is generally known, oral health care is expensive everywhere. We are creating a deep learning system to increase our capacity to identify oral cancer at an early stage.

2.3 Summary

This chapter provided the idea about the motivation towards our project which aims to detect oral cancer at an early stage using deep learning system.

CHAPTER 3:
RELATED WORKS

3.1 Introduction

This chapter discusses the many types of medical systems now available on the market. We also focus on the flaws in the existing system, and a proper argument will be offered as to why our approach is the best in the current situation.

3.2 Systems related to our project

In the paper "Computer-assisted Medical Image Classification for Early Diagnosis of Oral Cancer Using Deep Learning Algorithms," Jeyaraj, P. R., and Samuel Nadar, E. R., wrote that I conceived and built a deep learning method for an autonomous cancer diagnostic system based on a partitioned convolution neural network. They offer the examined aspects of hyperspectral medical pictures from oral cancer case studies, and they used the stochastic neighbor embedding technique to visually represent the assessed components of the hyperspectral images. They evaluate the performance of deep CNNs developed against that of SVM, DBN, and other classic classification techniques. This suggests that deep CNN can provide proper categorization after only one period of training. As a result, our deep learning algorithm may be simply implemented on a simple workstation to give an autonomous medical picture classifier without the need for professional experience.

John Gibson discussed how to take and analyze images of the oral cavi in the paper, "Automated Detection and Classification of Oral Lesions Using Deep Learning for Early Detection of Oral Cancer," by Roshan K, Alex Welikala, Paolo Remagnino, Jian Han Lim, Chee Seng Chan, Senthilmani Rajendran, Thomas George Kallarakkal, Rosnah Binti Zain, and Ruwan Duminda Jayasinghe The contribution of this paper is a unique way of combining bounding box annotations from many doctors, followed by a comparison of two deep learning-based automation systems. These positive first results show that deep learning is effective and capable of handling this difficult

task. Performance is projected to increase as the dataset develops, which will have a significant impact in low- and middle-income countries with limited health resources.

Rasheed Omobolaji Alabi, Alhadi Almangush, Mohammed Elmusrati, and Antti A. Mäkitie focused on deep learning's technical skills and methodologies for OSCC in their work, "Deep Machine Learning for Oral Cancer: From Precise Diagnosis to Precision Medicine." It investigates how deep learning may be used to detect cancer, categorize photographs, segment them, synthesize them, and plan treatments. Finally, they discuss how this technology can aid in precision medicine as well as the future of deep learning in the treatment of oral squamous cell carcinoma.

In the paper "Computer-assisted Medical Image Classification for Early Diagnosis of Oral Cancer Employing Deep Learning Algorithm Paper," Jeyaraj, P. R., and Samuel Nadar, E. R., They developed a new learning system that uses weighted majority voting on given criteria to categorize normal, pre-, and post-cancerous areas in hyperspectral image datasets. Models such as SVM, SVM-PCA, and DBM were designed and built, and the resultant classifiers were fused. In the beginning, they lowered the characteristics of the hyperspectral picture in the SVM and SVM PCA models. The suggested fusion approach improves mixed pixel detection sensitivity while outperforming existing deep learning methods in terms of accuracy.

3.3 Feasibility Study Indicating Possible Solutions

Deep learning will be used to construct our system. Deep learning is an artificial intelligence (AI) and machine learning approach influenced by human learning. Deep learning is a significant component of data science, which includes statistics and predictive modeling. Throughout the development of our system, we will use either the KNN or CNN model. In our model, we will use the VGG16 architecture. We will also employ deep pooling, flattening, dense, and fully linked layers, among other things. We will employ a confusion matrix for performance analysis, such as true positive, true negative, and so on. We will be utilizing kaggle for datasets. Our system will receive data as input, split it, process it, train it, and finally offer an acceptable outcome.

CHAPTER 4: PROCEDURE AND MATERIALS

4.1 Introduction

The information was obtained from publicly available internet-based sources. Following the separation of the preparation and test sets, the proposed method begins by stacking and eliminating photos and names from raw datasets, followed by preprocessing and expansion procedures. The proposed procedure's structure, as well as hyper-boundary organization, regularization procedures, and a streamlining calculation, are subsequently shown. Finally, estimates for network preparation and execution are provided.

4.2 Dataset Description

In the proposed system Histopathologic Oral Cancer Detection dataset has been used. [23] There are three different files for training data, validation data and test data. This System’s dataset contains 5192 Histopathologic images. 2494 of Normal Histopathologic images and 2698 images of oral cancer effected Histopathologic images were used on this system. Each folder contains two classes- Normal and OSCC. Figures 1 and 2 show the OSCC and normal image samples from the dataset.

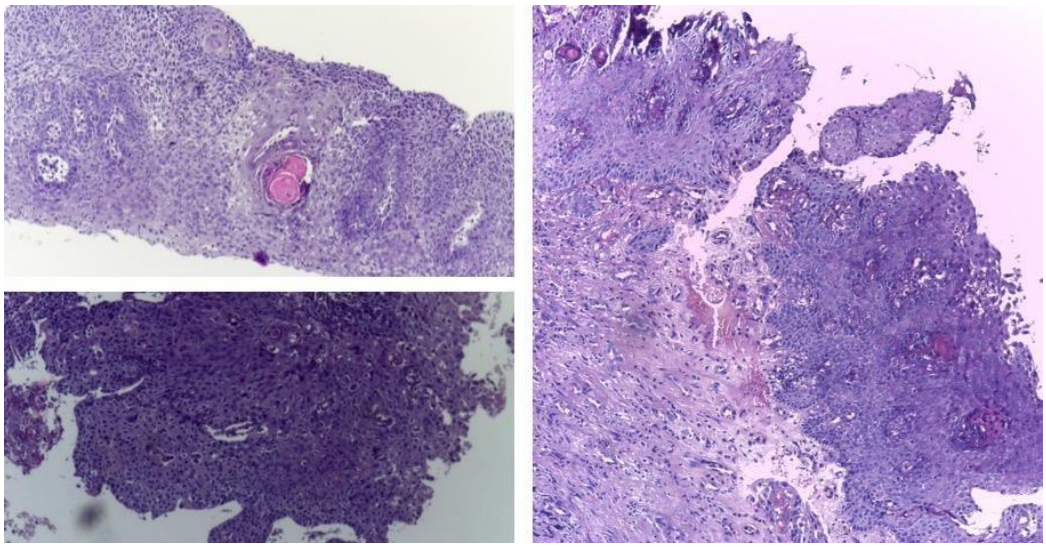


Figure 1: Images of OSCC

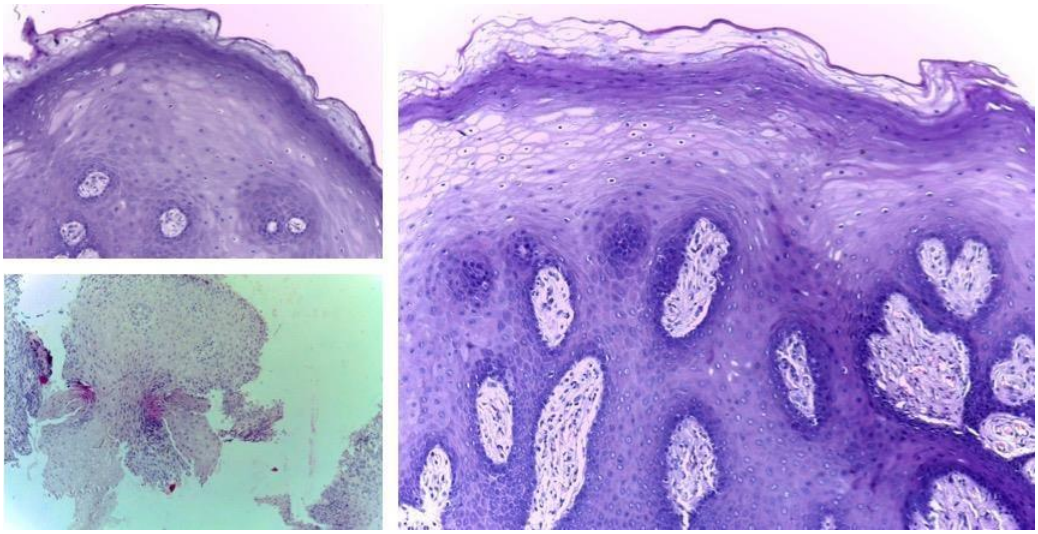


Figure 2: Images of Normal

4.2.1 Data Preprocessing

A preprocessing step is conducted prior to feeding the picture into the proposed structure. The first step is to reduce the size of the original image from 512*512 to 128*128 pixels. Simpler computations are utilized to reduce the dimensionality and complexity of the network, as well as to assist the network in performing better in less time. The data is jumbled before splitting to preserve the system's capacity to train on unsorted data and to avoid the system focusing primarily on a small portion of the whole dataset.

4.2.2 Data Augmentation

Images of various sizes were obtained from the database and saved in the int16 format. Because they represent the network's input layer, these photos have been normalized and reduced to 128 128 pixels. We enhanced the images under consideration so that the system would see them as fresh,

which is a typical method for avoiding overfitting and improving model resilience. We altered each image in various ways to contribute to the dataset. To begin, rotate the picture by 30 degrees. We mirrored it from right to left, reversed it, and moved it 0.1 inch in width and height to achieve this effect.

4.3 Block Diagram

Figure 3 depicts a block diagram outlining the research procedure.

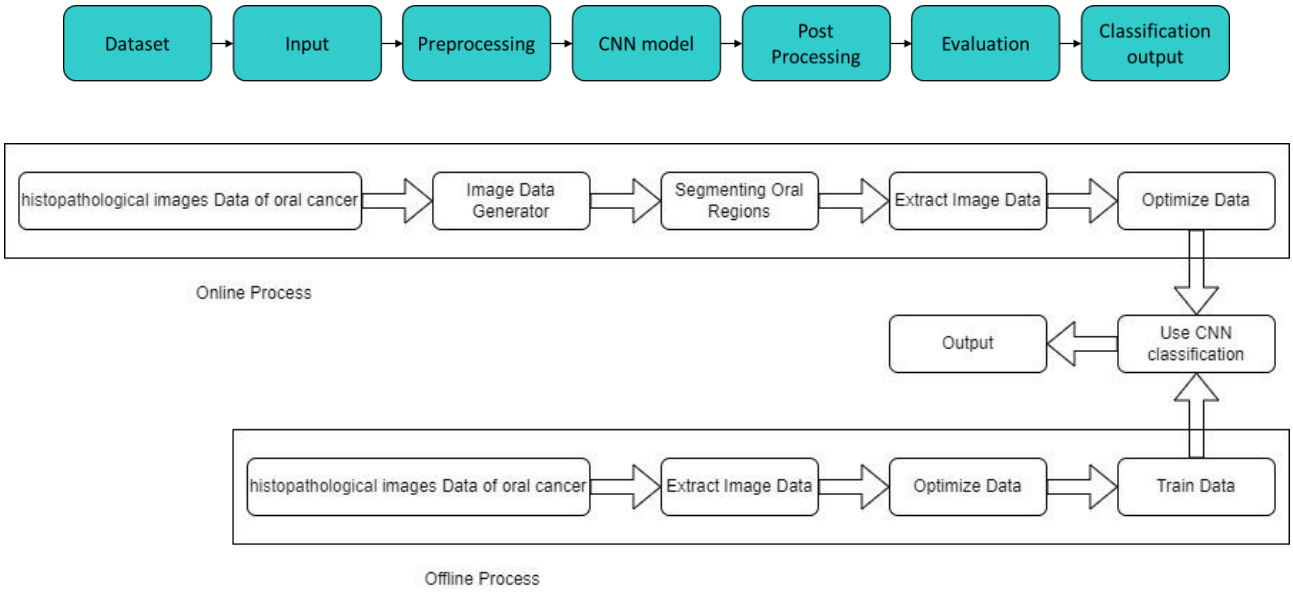


Figure 3: System Block Diagram

We divided the process in two categories one in on online and another is on offline one the online process we choose google colab to train our data with a custom CNN model and on the offline process we chose our local device with a combination of GPU so that our process can be done with a short time. Both the processes are same. The collection addresses two OSCC and two Normal courses. Changing the dimensionality of photographs and employing information expansion

techniques such as rotating, flipping, zooming, and cutting are all critical for the preprocessing system. Following preprocessing, the images are sent into the bespoke CNN model. As a result, post-handling procedures such as K-crease cross-approval and boundary calibrating were used to improve preparation precision. The constructed model is then tested against the test dataset's cryptic pictures, and a disarray grid is created to demonstrate the model's responsiveness. Finally, an image of a patient is shown to the model in order to determine its correct kind and confirm its reliability. The block graph depicts the complete framework in the simplest structure possible. The proposed technique's framework engineering is displayed in Figure 4.

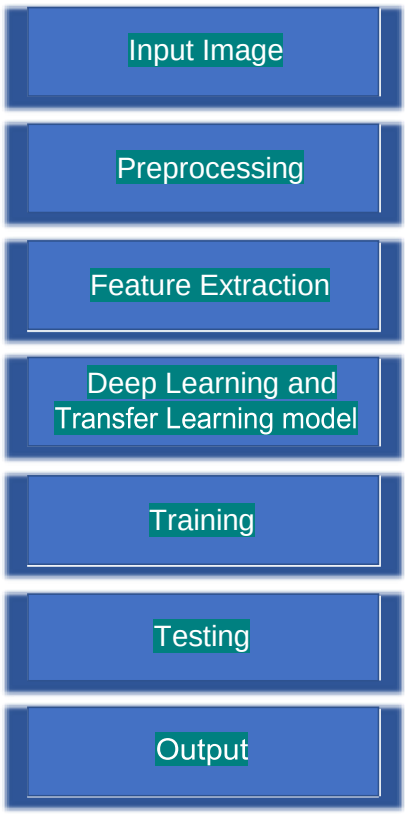


Figure 4: System Architecture of the Proposed Method

Figure 4 displays the various methodologies employed in this study to classify colon cancer. One way is to divide the data into training and testing sets and use a tenfold cross-validation procedure. To train a model using another technique, divide the training data into training, validation, and test subsets. After that, the model may be trained without utilizing K-fold cross-validation.

4.4 Proposed System

In this network's convolution block, we used convolutional layers, ReLu, and Dropout. Convolutional layers behave similarly to max-pooling layers. The softmax function is found in the last layer. A 2D convolutional layer moves K convolutional filters (kernels) of size (M X N) over the input images, computing the dot product of the weights (kernel weights) and the input. Stride (S) refers to how the filters move vertically and horizontally over the image. To keep information at the borders, padding (P) of the original photographs may be applied prior to sliding the filters. Dropout (D) is a simple method for avoiding neural networks from overfitting. These kernels are used to find features, ranging from early-layer kernels that just seek low-level features like edges, lines, and blobs to more sophisticated kernels that hunt for more complicated features. For the convolutional network, we used the following parameters: filter = 64, 128 and 128, kernel (M* N) = 3, 3, 3, and 3, Stride = [1, 1, 1], Padding = same, Dropout = 0.2. We determined that a dropout probability of 20% was best suited for dropout layers. Every convolutional layer is followed by a non-saturated activation function known as ReLU, which is primarily used to significantly shorten training time when compared to other activation functions. The ReLU model is described as a function of x, with the output equaling the input when x is positive and 0 for all other values. The ReLU function formula is:

$$f(x) = max(0, x) \quad (1)$$

Maximum pooling is a type of down sampling that is used to produce spatial invariance by splitting the entire image into tiny rectangles (3x3 in the recommended structure) that traverse over the image in a predetermined step (3x3) and then analyzing only the maximum value of the four components. The pooling layer is used to reduce the number of parameters in the network and hence the number of computations.

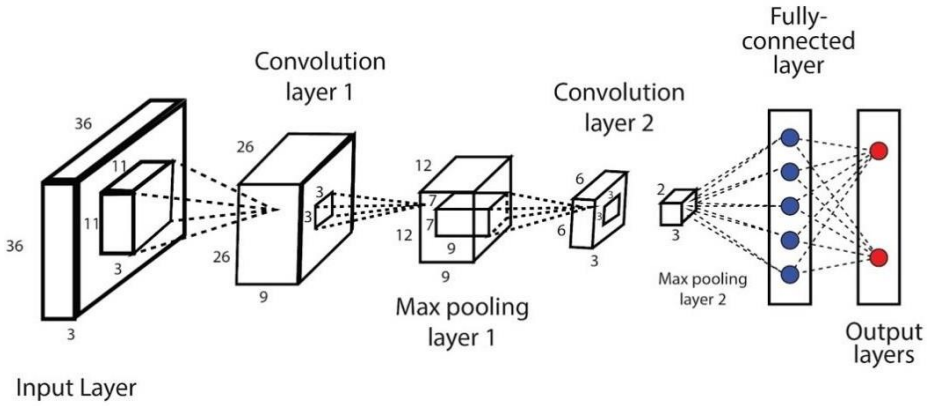


Figure 5: Proposed CNN Architecture

As shown in Figure 5, the network design consists of an input, three convolutional blocks, a classification block, and an output. The first significant block, block 1, is made up of a convolutional layer that produces an output that is twice the size of the input. The first convolution layer is convolved using 3*3*64 filters with stride = 1 and padding equal to the value of the first convolution layer. Following convolution, the convolutional layer behaves similarly to the rectified linear unit (ReLU) activation layer and the dropout layer as a 20% dropout. This block also includes a max-pooling layer, which produces an output that is twice as small as the input. The second block configuration, block 2, takes the output of Block 1 as input and convolves it using 3*3*128 filters with stride = 1 and padding = 1, but no dropout layer is used. Rather than employing a dropout layer, we used batch normalization, which allows each layer of the network to be independent. Train on your own. The response of the convolutional layer is controlled by the rectified linear unit (ReLU) activation layer and the maximum pooling layer. This block produces an output that is twice the size of the input after max-pooling the convolution layer's response. The third block, block 3, convolution setup, is identical to Block 2, except that we used a 20% dropout this time. The maximum pooling layer and the rectified linear unit (ReLU) activation layer define the response of

the convolutional layer. This block takes the output of block 2 as an input and, after max-pooling the convolution layer's response, produces an output that is twice the size of the input. The output of this block is used as the input for the next block, which is the classification block. The classification block is made up of four fully connected (FC) layers, the first of which shows the flattened output of the previous max-pooling layer to 8192 nodes. The second layer has 128 nodes, whereas the third layer contains 64 nodes. Softmax was utilized in the final thick layer, which contains as many hidden units as there are colon cancer categories.

4.5 Cross-Validation

In this study, the performance was evaluated using the cross-validation method. We may gather more metrics and draw important conclusions about our algorithm and data. To begin, the entire data set is divided into two halves, with 90% of the data being training images and the other half containing test images. We just used the training data for cross-validation. As shown in Figure 6, data is often divided into ten nearly equal pieces, nine of which can be utilized for training and one for validation. Each part has an equal amount of data, which assists in proper cross-validation. In each iteration, nine randomly chosen sections are utilized to train the model, while another piece is used to validate the model. Test data was kept separate from production data from the start, so we could look at the model with photos that we didn't know about.

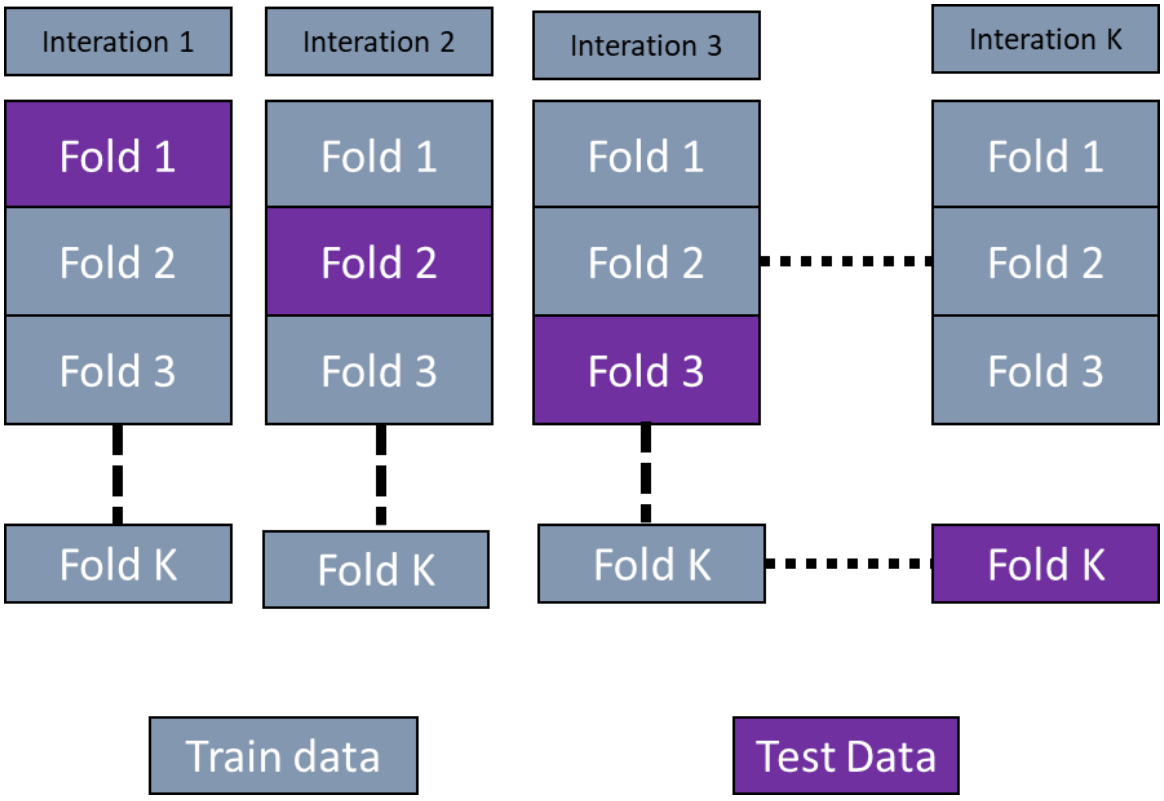


Figure 6: K- fold cross-validation

4.6 Performance Matrix

The system displayed real and expected values on a confusion matrix. In a classification model, the confusion matrix describes the prediction outcomes. The following values have been determined for precision, sensitivity, specificity, and accuracy:

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (4)$$

$$F1-Score = 2*\frac{Precision*Recall}{Precision+Recall} \quad (5)$$

Whereas "True Positive" refers to the number of situations that were expected to be favorable that actually occurred (TP). True Negative (TN) is the percentage of projected negative occurrences that are also true negative. A False Negative (FN) is the number of projected negative occurrences that turn out to be positive, also known as a type two error. A "False Positive" is the number of projected positive cases that turn out to be negative (FP).

CHAPTER 5: RESULTS AND ANALYSIS

5.1 Introduction

In this chapter, we used a custom CNN model in our proposed system. The data was utilized to train the model as well as to perform 10-fold cross-validation on its performance. The model's training accuracy was 100% and its validation accuracy was 99% without cross-validation. After evaluation, the accuracy of the test was confirmed to be 99 percent. Cross-validation was utilized to tenfold the model's performance. The test's accuracy after cross-validation was 99.33 percent. The model's performance was improved by using cross-validation.

5.2 Model Accuracy and loss

Figure 7 and 8 show the training and validation accuracy and training and validation loss graph, respectively. With a learning rate of 0.0001, 64 batch sizes, and 200 epochs, the model was trained. The model's performance improves with each epoch. In the first few epochs, the performance improves dramatically. After 5 epochs, the improvement slows down, and after 25 epochs, the performance barely improves.

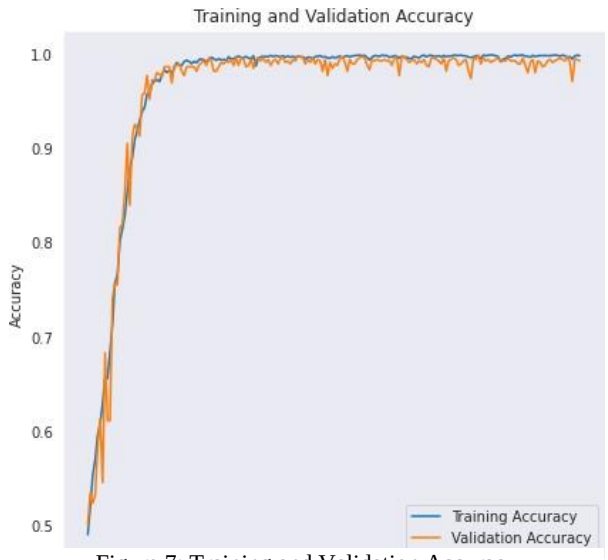


Figure 7: Training and Validation Accuracy

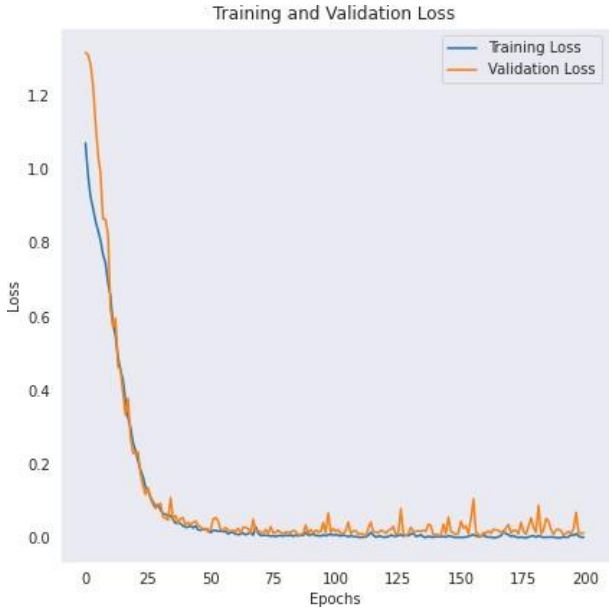


Figure 8: Training and Validation Loss

Figure 8 shows that training accuracy is 100% and validation accuracy is 99.33%. Also, in figure 8, validation loss is higher than training loss. Where with the increasing of number of epochs the loss decreased. And after 50 epochs is stays constant.

As the training accuracy is higher than validation accuracy and validation loss is higher than training loss so this indicates that our proposed model is not over- fitted.

5.3 Model Evaluation

The network's performance was examined using 10-fold cross-validation, and the network's improvement was measured using augmented pictures. The suggested network performs substantially better, with an average accuracy of 99.33 percent on average. Without cross-validation, the model's accuracy is 99 percent. Furthermore, with an average accuracy of 99.33 percent, the record-wise cross-validation method produced the best results for the 10-fold cross-validation method. This means that the model can be used to accurately identify a number of different types of colon tissue. Figure 9 shows the classification report of the proposed algorithm.

Classification Report

	precision	recall	f1-score	support
0	0.96	0.99	0.97	71
1	0.99	0.98	0.99	136
accuracy			0.98	207
macro avg	0.98	0.98	0.98	207
weighted avg	0.98	0.98	0.98	207

Figure 9: Classification Report

Figure 10 below describes the performance of a classification model on a set of test data where the true values are known. Here the 00 and 11 level shows the true result and 01 and 10 shows the false result of our model. True result was 210 and false result was 4.

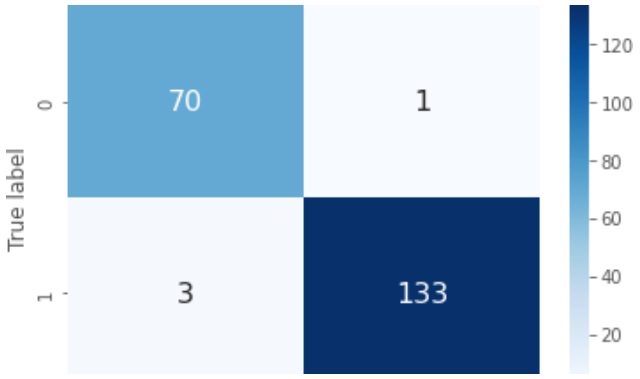
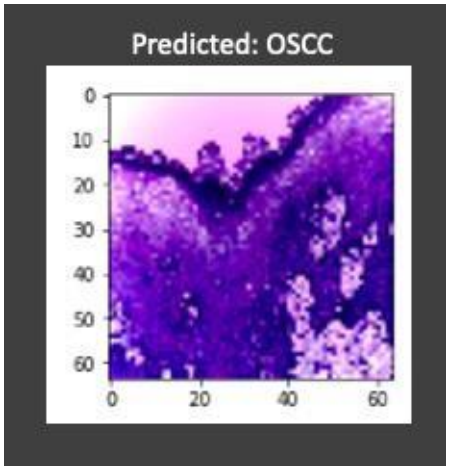
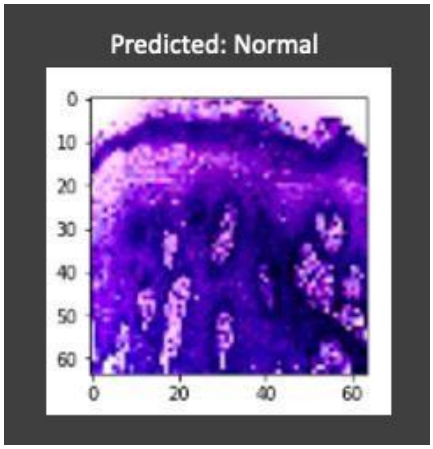


Figure 10: Confusion matrix of classification

5.4 Model Test

Our research is primarily concerned with real-world issues that may aid in the classification of oral cancer. This research used Histopathological images to classify oral cancer. A few data points were extracted from the main dataset to evaluate the model's classification accuracy.

Table 1: Model Test Result

Input Type	Predicted Type
OSCC	
Normal	

In the preceding table, a real-world picture is used to test the model, and the trained model accurately predicts when given a random image. The model is capable of differentiating between two types of oral images.

5.5 Comparison of result

Two methodologies were used in this work to test the suggested CNN model. Table 2 compares the proposed CNN model to research that has used bespoke CNN designs. Without 10-fold cross validation, the suggested model was 93 percent accurate. This is superior than current systems.

Table 2: Model's Accuracy Comparison

Reference	Algorithm	Accuracy (%)
This paper	**Custom CNN**	**99.33**
Ref [10]	ANN	86.9
Ref [11]	ResNet50	80.0
Ref [12]	CNN	89.2

Here, by using a custom CNN model, this paper achieved 99.33% accuracy, but using CNN model ref [12], it achieved 89.2% accuracy. Also, using a pre-trained model such as ResNet50, ref [11] achieved 80 percent accuracy.

5.6 Summary

This chapter, we used a custom CNN model in this proposed system. The data was used to train the model as well as to perform 10-fold cross-validation on its performance. The model's training accuracy was 100% and its validation accuracy was 99% without cross-validation. The accuracy of the test was determined to be 99 percent after examination. Cross-validation was used to tenfold the model's performance. The accuracy of the test after cross-validation was 99.33 percent.

CHAPTER 6: SKILLS

6.1 Introduction

In this chapter we discuss the skills that we have obtained in order to develop this massive sophisticated system.

6.2 Skills obtained

Through this project the following skills have been developed:

- **Skill in Programming & Tools**
  - **Python**

    Python is a computer programming language that is frequently used to create websites and applications, automate operations, and analyze data. Python is a general-purpose programming language, which means it can be used to develop a wide range of applications and is not specialized for any particular problem.

  - **Google Colab**

    Google Colaboratory, or "Colab" for short, is a Google Research product. Colab allows anybody to develop and run arbitrary Python code in the browser, making it ideal for machine learning, data analysis, and teaching. In more technical terms, Colab is a hosted Jupyter notebook service that requires no setup and provides free access to computer resources such as GPUs.

6.3 Summary

In this chapter, we addressed the abilities that were acquired during the process of building and materializing this system.

CHAPTER 7: DESIGN IMPACT

7.1 Introduction

In this chapter, we discuss about the various impacts that our system has been able to generate.

7.2 Environmental Impact

By using this system at a diagnostic center, vast amounts of paper may be saved since all jobs will be digitized, reducing the quantity of paper wasted every day in official work.

7.3 Economic Impact

The economic benefit of this system is that people can detect cancer at an early stage and treat it at a lower cost.

7.4 Social Impact

Our system will be socially acceptable since this type of system is urgently needed. In this day and age, everything has been automated to make life easier for people. As a result, our system is no exception.

7.5 Sustainability

Our system is capable of handling a large number of patients' information at the same time. When a large number of tests are performed at the same time, our system remains steady. As a result of these facts and ongoing testing, our system is sustainable.

7.6 Summary

This chapter has detailed and discussed the many forms of impacts available through our system. Based on the effects listed above, we may infer that our created system is suitable for usage in any situation.

CHAPTER 8: COMPLIANCE WITH IEEE STANDARDS

8.1 Introduction

This section discusses the consistency of our assignment with several criteria. There are several various standards, among which the IEEE standards, US standards, and European standards are discussed in this section.

8.2 Compliance with IEEE standard

IEEE Standards affiliation has proposed a few separate guidelines. The vast majority of them, however, are not relevant to our approach. We have integrated the IEEE standard operating concept.

8.3 Compliance with US standard

ANSI advises that copyrighted software be supplied solely for informative reasons or in formats that do not require specific implementations of the standard. Object code should never be introduced as a normative requirement in a standard. While ANSI opposes the use of software standards to mandate specific implementations and believes that the use of software in standards should be avoided to the greatest extent possible, ANSI acknowledges that there may be circumstances in which the inclusion of some software, if accompanied by adequate legal permission, may facilitate the development of multiple, competing, and interoperable implementations of the standard. Some examples of such software include:

- Schema examples;
- Data structure definitions;
- ASN.1 structure definition;
- ABNF grammar specifications;

- Example programming instructions that are sufficiently constrained in scope that they do not perform a complete or significant part of a function, either singly or in aggregate, and are illustrative, at best, of limited sections of a fully specified independent specification; or
- Sample programming instructions provided solely for conformance testing purposes.

Our project has been established based on the above ANSI principles and it completely relies upon it.

8.4 Summary

In this section, we analyzed the many complying standards and ensured that we are in compliance. These criteria have been set in place without hesitation in order to manage things, assure well-being, and ensure there are no health risks associated with the usage of various parts. Maintaining these measures is critical, and we have done so over the course of our job work.

CHAPTER 9: CONCLUSION

This work introduced a novel CNN architecture for colon tissue categorization. The categorization was carried out utilizing a database of T1-weighted, contrast-enhanced histopathological images, including two distinct oral types. Because we utilized full photos as input, no preprocessing or segmentation of the oral cancer was required. Our built-in neural network is more straightforward than pre-trained networks, and it can be operated on standard current desktop PCs. This is achievable since the method needs less training and implementation resources. The relevance of establishing smaller networks is also related to the algorithm's portability, which is critical for diagnostics in underdeveloped nations. Without using cross-validation, the accuracy was 93%. On both the original and enriched picture datasets, we performed 10-fold cross-validation record-by-record and subject-by-subject. When compared to similar state-of-the-art approaches, our network outperformed them. Record-wise cross-validation with a more complete set of data was the best way to get the best results for 10-fold cross-validation, with an accuracy of 99.33 percent.

BIBLIOGRAPHY

[1] Oral cancer, https://en.wikipedia.org/wiki/Oral_cancer

[2] Oral Leukplakia LM328"https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSM65276

[3] Oral cancer, https://www.webmd.com/oral-health/guide /oral-cancer#1

[4] Head-and-Neck-squamous-cellcarcinoma, https://en.wikipedia.org/wiki/Head_and_neck_ squamous_cell_carcinom, Wikipedia(2018).

[5] Squamous-cell-carcinomacancer,https://en.wikipedia.org/wiki/Head_and_neck_squamouscell Carcinoma

[6] Staging of Lip and Oral Cavity Cancer , https://www.ahns.info/patient-information/

[7] Fatihah Mohd, Noor Maizura Mohamad Noor, Zainab Abu Bakar, Zainul Ahmad Rajion, Analysis of Oral Cancer Prediction using Features Selection with Machine Learning, ICIT 2015 The 7th International Conference on Information Technology.

[8] Ahmad LG*, Eshlaghy AT, Poorebrahimi A, Ebrahimi M and Razavi AR, Using Three Machine Learning Techniques for Predicting Breast Cancer Recurrence, Health & Medical Informatics (2013), 2157-7420.

[9] Harikumar Rajaguru and Sunil Kumar Prabhakar, Performance Comparison of Oral Cancer Classification with Gaussian MixtureMeasures and Multi Layer Perceptron, The 16th International Conference on Biomedical Engineering p(2017) 123-129.

[10] Amy F. Ziober,Kirtesh R. Patel,Faizan Alawi, Phyllis Gimotty,4 Randall S. Weber, MichaelM. Feldman,Ara A. Chalian,Gregory S. Weinstein,Jennifer Hunt, and Barry L. Ziober, Identification of a Gene Signature for Rapid Screening of Oral Squamous Cell Carcinoma, American Association for Cancer (2018).

[11] Marc Aubreville, Christian Knipfer, Nicolai Oetter, Christian Jaremenko, Erik Rodner, Joachim Denzler, Christopher Bohr, Helmut Neumann, Florian Stelzle, & Andreas Maier,

Automatic Classification of Cancerous Tissue in Laserendomicroscopy Images of the Oral Cavity using Deep Learning, SCIENTIFIC Reports,(2017), 7: 11979.

[12] Symtoms of oral cancer, https://www.ahns.info/resources/education/patient_education /oralcavity

[13] Martin Halicek, Guolan Lu, James V. Little, Xu Wang, Mihir Patel, Christopher C. Griffith, Mark W. El-Deiry, Amy Y. Chen, Baowei Fei, Deep convolutional neural networks for classifying head and neck cancer using hyperspectral imaging, J. Biomed.Opt. 22(6), 060503 (2017), doi: 10.1117/1.JBO.22.6.060503.

[14] Randomforest, https://towardsdatascience.com/therandom-forest-algorithm-d457d499ffcd

[15] Konstantina Kourou , Themis P. Exarchos , Konstantinos P. Exarchos ,Michalis V. Karamouzis, Dimitrios I. Fotiadis, Machine learning applications in cancer prognosis and prediction, Computational and structural biotechnology journal, (2015),18-17.

[16] Wafaa K. Shams and Zaw Z. Htike, Oral Cancer Prediction Using Gene Expression Profiling and Machine Learning, International Journal of Applied Engineering Research, (2017) ,0973-4562.

[17] Heba M, El-Dahshan ESA, El-Horbaty ESM, Abdel-Badeeh M (2018) Classification using deep learning neural networks for brain tumors. Future Comput Inf J 3(1):68–71.

[18] Deepak Kumar J, Surendra Bilouhan D, Kumar RC (2018) An approach for hyperspectral image classification by optimizing SVM using self-organizing map. J Comput Sci 25(1):252–259.

[19] Yuan Y, Lin J, Wang Q (2015) Hyperspectral image classification via multitask joint sparse representation and stepwise MRF optimization. IEEE Trans Cybern 46(12):2966– 2977.

[20] Dou Q, Chen H, Yu L et al (2016) Automatic detection of cerebral microbleeds from MR images via 3D convolutional neural networks. IEEE Trans Med Imaging 35(5):1182–1195.

[21] Wang C, Gong L, Yu Q et al (2017) DLAU: a scalable deep learning accelerator unit on FPGA. IEEE Trans Computer Aided Des Integr Circuits System 36(3):513–517.

[22] Kalantari N, Ramamoorthi R (2017) Deep high dynamic range imaging of dynamic scenes. ACM Trans Graph 36(4):1–12.

[23] Oral Cancer Data. https://www.kaggle.com/code/hossamfakher/classification-with-transfer-learning/data

APPENDIX

Software: Google Colab

Language: Python

Code:

```
!pip install split-folders

from google.colab import drive
drive.mount('/content/drive')

!pip install scikit-plot

# import all libraries
# Train/Test
Libraries

import os
import numpy as np
import tensorflow as tf
from collections import Counter
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix
import scikitplot
from sklearn.metrics import roc_curve, auc
from sklearn.metrics import classification_report
```

```
#data preprocessing
data_aug_train = tf.keras.preprocessing.image.ImageDataGenerator(
   height_shift_range = 0.15,
   width_shift_range = 0.15,
   rotation_range = 10,
   shear_range = 0.1,
   fill_mode = 'nearest',
   zoom_range = 0.2)

train_generator = data_aug_train.flow_from_directory(
       '/content/splitted_data/train',
       target_size=(224, 224),
       batch_size = 32,
       class_mode ='categorical',
       color_mode = 'rgb',
       classes = ['OSS',
       'healthy'], seed = 2,
       shuffle = True,
       interpolation = 'lanczos'
       )

data_aug_val = tf.keras.preprocessing.image.ImageDataGenerator()

val_generator = data_aug_val.flow_from_directory(
       '/content/splitted_data/val',
       target_size =(224, 224),
       batch_size =32,
       class_mode ='categorical',
```

```
    color_mode = 'rgb',
    classes= ['OSS',
    'healthy'], seed = 2,
    shuffle = True,
    interpolation = 'lanczos'
    )

data_aug_test = tf.keras.preprocessing.image.ImageDataGenerator()

test_generator = data_aug_test.flow_from_directory(
    '/content/splitted_data/test',
    target_size = (224, 224),
    batch_size = 207,
    class_mode ='categorical',
    color_mode = 'rgb',
    classes = ['OSS',
    'healthy'], shuffle = False,
    interpolation = 'lanczos'
    )

train_generator.class_indices, val_generator.class_indices

# Using pretrained MobileNetV2 model

def MobileNetV2_Model():

  baseModel = tf.keras.applications.MobileNetV2(weights="imagenet", include_top=False, input_shape=
(224, 224, 3))
```

```
 output = baseModel.output
 output = tf.keras.layers.GlobalAveragePooling2D()(output)
 output = tf.keras.layers.Dense(1024, activation="relu")(output)
 output = tf.keras.layers.Dropout(0.15)(output)
 output = tf.keras.layers.Dense(512, activation="relu")(output)
 output = tf.keras.layers.Dropout(0.15)(output)
 output = tf.keras.layers.Dense(2, activation="softmax")(output)
 model = tf.keras.Model(inputs=baseModel.input, outputs=output)
 for layer in baseModel.layers:
   layer.trainable = False
 return model


model = MobileNetV2_Model()


print("[INFO] compiling
model...") INIT_LR = 0.001
EPOCHS = 10
BATCHSIZE = 64
optimizer = tf.keras.optimizers.Adam(lr=INIT_LR, decay=INIT_LR / EPOCHS)
model.compile(loss="categorical_crossentropy", optimizer=optimizer, metrics=[tf.keras.metrics.Categori
calAccuracy(), tf.keras.metrics.AUC()])


print(model.summary())


modelPath = '/content/drive/MyDrive/Training/Oral_cancer/saved_models_final2/Pretrained MobileNe
tV2'
if not os.path.exists(modelPath):
```

```
  os.makedirs(modelPath)

print('Model Directory Created')

else:

  print('Model Directory Already Exists')


reduceLROnPlat = tf.keras.callbacks.ReduceLROnPlateau(monitor='val_categorical_accuracy',
 factor=0.8, patience=10, verbose=1, mode='max',

                                        min_delta=0.0001, cooldown=5, min_lr=0.0001)

early = tf.keras.callbacks.EarlyStopping(monitor="val_categorical_accuracy", mode="max", patience=8)


model_checkpoint = tf.keras.callbacks.ModelCheckpoint(modelPath+'/MobileNetV2-best-
model.h5', monitor='val_categorical_accuracy',

                                        verbose=1, save_best_only=True, mode='max')


STEP_TRAIN = len(train_generator)


modelHistory = model.fit(train_generator, steps_per_epoch=STEP_TRAIN,

                    validation_data= val_generator, epochs=EPOCHS, verbose=1,
callbacks=[model_checkpoin t, reduceLROnPlat, early])


tf.keras.models.save_model(model, modelPath+'/MobileNetV2-
model.h5', overwrite=True, include_optimizer=True, save_format=None,

                     signatures=None, options=None)

import pickle

with open('/content/drive/MyDrive/Training/Oral_cancer/saved_models_final2/Pretrained MobileNetV
2/MobileNetV2_trainHistoryDict', 'wb') as file_pi:

  pickle.dump(modelHistory.history, file_pi)


import matplotlib.pyplot as plt

plt.style.use('ggplot')
```

```
def plot_history(history):
    acc = history['categorical_accuracy']
    val_acc =
    history['val_categorical_accuracy'] loss =
    history['loss']
    val_loss =
    history['val_loss'] auc =
    history['auc']
    val_auc =
    history['val_auc'] x =
    range(1, len(acc) + 1)

    plt.figure(figsize=(18, 5))
    plt.subplot(1, 3, 1)
    plt.plot(x, acc, 'b', label='Training acc',marker = 'p',color='green')
    plt.plot(x,val_acc, 'r', label='Validation acc',marker = 'p',color='red')
    plt.title('Training and Validation accuracy')
    plt.xlabel('Epoch')
    plt.ylabel('Accuracy')
    plt.legend(loc='lower right' )
    plt.subplot(1, 3, 2)
    plt.plot(x, loss, label='Training loss',marker = 'p',color='green')
    plt.plot(x, val_loss, label='Validation loss',marker = 'p',color='red')
    plt.title('Training and Validation loss')
    plt.xlabel('Epoch')
    plt.ylabel('Loss')
    plt.legend()
    # plt.savefig('curve.jpg',dpi=600)
    plt.subplot(1, 3, 3)
```

```
plt.plot(x, auc, 'b', label='Training loss',marker = 'p',color='green')
```

```
    plt.plot(x, val_auc, 'r', label='Validation loss',marker = 'p',color='red')
    plt.title('Training and Validation AUC')
    plt.xlabel('Epoch')
    plt.ylabel('AUC')
    plt.legend()
    plt.savefig('curve.jpg',dpi=600)


# plot_history(modelHistory.history)


!cp -r '/content/drive/MyDrive/Training/Oral_cancer' -
d '/content/drive/MyDrive/Training/Oral_cancer/saved_models_final2'


!cp -r "/content/drive/MyDrive/Training/Oral_cancer/saved_models_final2" -d '/content'


# Evaluate the Best Saved Model
model = tf.keras.models.load_model('/content/drive/MyDrive/Training/Oral_cancer/saved_models_fina
l2/Pretrained MobileNetV2/MobileNetV2-best-model.h5')
loss, accuracy, auc= model.evaluate(test_generator, verbose=1)
print('Model Accuracy: {:0.2f} | Model Loss: {:0.4f} | Model AUC: {:.02f}'.format(accuracy, loss, auc))


batch=0
label =
[]
for x,y in test_generator:
  # cc.append(y)
  if batch==1:
  break
  label.append(y)
  batch+=1
```

```
y_true = np.array([np.where(label[0][i] == 1)[0][0] for i in range(len(test_generator.filenames))])
y_true.shape

Y_pred = model.predict_generator(test_generator, 1)
y_pred = np.argmax(Y_pred, axis=1)

classes= ['0', '1']
print('Classification Report')
print(classification_report(y_true, y_pred, target_names=classes))

import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

def make_confusion_matrix(cf,
                          group_names=None,
                          categories='auto',
                          count=True,
                          percent=True,
                          cbar=True,
                          xyticks=True,
                          xyplotlabels=True,
                          sum_stats=True,
                          figsize=None,
                          cmap='Blues',
                          title=None):

    blanks = ['' for i in range(cf.size)]
```

```
    if group_names and len(group_names)==cf.size:
        group_labels = ["{}\n".format(value) for value in group_names]
    else:
        group_labels = blanks


    if count:
        group_counts = ["{0:0.0f}\n".format(value) for value in
    cf.flatten()] else:
        group_counts = blanks


    if percent:
        group_percentages = ["{0:.2%}".format(value) for value in
    cf.flatten()/np.sum(cf)] else:
        group_percentages = blanks


    box_labels = [f"{v1}{v2}{v3}".strip() for v1, v2, v3 in
    zip(group_labels,group_counts,group_percentages
)]
    box_labels = np.asarray(box_labels).reshape(cf.shape[0],cf.shape[1])


    if sum_stats:
        accuracy = np.trace(cf) / float(np.sum(cf))
        if len(cf)==2:
            precision = cf[1,1] /
            sum(cf[:,1]) recall = cf[1,1] /
            sum(cf[1,:])
            f1_score = 2*precision*recall / (precision + recall)
            stats_text = "\n\nAccuracy={:0.3f}\nPrecision={:0.3f}\nRecall={:0.3f}\nF1
                Score={:0.3f}".format( accuracy,precision,recall,f1_score)
```

```
        else:
            stats_text =
    "\n\nAccuracy={:0.3f}".format(accuracy) else:
        stats_text = ""

    if figsize==None:
        figsize = plt.rcParams.get('figure.figsize')

    if xyticks==False:
        categories=False

    plt.figure(figsize=figsize)
    sns.heatmap(cf,annot=box_labels,fmt="",cmap=cmap,cbar=cbar,annot_kws={'size':16},xticklabels=cat
egories,yticklabels=categories)

    if xyplotlabels:
        plt.ylabel('True label')
        plt.xlabel('Predicted label' + stats_text)
        plt.savefig('conf_mat.jpeg',dpi=600)
    else:
        plt.xlabel(stats_text)
        plt.savefig('conf_mat.jpeg',dpi=600)

    if title:
        plt.title(title)
        plt.savefig('conf_mat.jpeg',dpi=600)

conf_mat=confusion_matrix(y_true, y_pred)
```

```
make_confusion_matrix(conf_mat, figsize=(6,4),percent=False,sum_stats=False,categories=['0', '1'],cbar
=True,cmap='Blues')


from sklearn.metrics import cohen_kappa_score
```