Innovations

Artificial Intelligence for Human Disease Prediction

Shweta Marulkar [1], Bhavana Narain[2]& Rajiv kumar Mente [3]

[1&2]MATS University(Raipur)

[3]PAH Solapur University(Solapur)

Corresponding author: **Shweta Marulkar**

Abstract

Since human nail disease does not exhibit clinical symptoms that are hazardous to one's health, it is typically overlooked. On the other hand, nail illness might be a precursor to a health problem. Some nail conditions can result in an infection, damage, or even the loss of the nail itself. It might diminish someone's visual appeal and attractiveness. Because nail diseases may take many different forms and share many characteristics, diagnosing nail diseases can be challenging for physicians. As a result, this study developed an automated technique for classifying nail diseases using nail images. The suggested approach was built using an Adam optimizer and the VGG-16 neural network architecture. In this study, nail conditions such as Koilonychia, Beau's Lines, and Leukonychia etc. were categorized. Python programming is used to simulate the model used in this study. According to the simulation findings model produced a classification accuracy of 94.27 %. The clinical diagnosis of nail disease is anticipated to be supported by the transfer learning technique based on a neural network simulation used in this study.

Keywords: *Artificial intelligence, Medical Image Processing, Nail image, CNN, VGG16, Human Disease.*

1. Introduction

Human nail images have the potential to provide valuable diagnostic information for various diseases. Several studies [11] have documented the utilization of image processing techniques for diagnosing diseases through nail. Studies [9], [10] have presented nail image-based biometric systems. In various research [11], the application of image processing techniques for diagnosing nail diseases has been explored. Maniyan and Shivakumar's survey suggested a nail disease classification approach based on color, texture, and shape analysis, utilizing a support vector machine as the classifier [12]. Similarly, Suguna et al. presented a method employing wavelet analysis and extracting color, texture, and shape features for nail disease classification [13]. While achieving reasonably high accuracy, these approaches face limitations when dealing with diverse image data and backgrounds. To overcome these limitations, a deep learning approach involving convolutional neural networks (CNNs) is proposed [14]. Hence, this study introduces an automated nail disease classification system using a CNN based on the VGG-16 architecture. The paper is structured into three main sections. Section 2 represents the review of literature. Section 3 outlines the study's materials and methods for classifying diseases. It delves into aspects such as dataset specifics, preprocessing methods, augmentation, modeling, and evaluation techniques. In Section 4, the paper presents the experiments, results, discussions, and evaluations of the proposed approach. Finally, Section 5 offers a conclusion summarizing key findings and insights drawn from the study.

2. Literature Review

Numerous researches have been undertaken and published in the literature to categorize fingernail illnesses. Using an image dataset made up of eight distinct kinds of anomalies, Banu and Devi et al. conducted a comparative investigation of several classifiers, including KNN and SVM. Indi and Patil et al. used, KNN, ANN and SVM classifiers to analyse the colour, shape, and characteristics of fingernails in order to predict illnesses related to them.

Deep learning methods have gained significant traction in diverse fields of research in AI, aiming to mechanize problem-solving procedures as well as improve decision making processes [3][4][5].

Over the past few years, the utilization of deep learning techniques has witnessed a surge in biomedical research, particularly in the field of Health assessments. The focus of research in medical diagnostics revolves around advancing healthcare over the utilization of medical imaging technology and the analysis of medical data [6][7][8].

Deep learning-based medical diagnostics can be used in a variety of biomedical images, including ultrasound, X-rays, MRI and other clinical images. The detection and study of illnesses to determine their presence, nature, and severity are included in the discipline of clinical diagnostic research [9].

EfficientNet has emerged as novel convolutional neural network architecture with a wide range of applications in image analysis. It has gained popularity as a preferred prediction model in various disease diagnostic studies involving humans as well as plants [10].

The popularity of EfficientNet among researchers in the field of deep learning stems from its remarkable performance in classification and prediction tasks. Numerous studies have compared the performance of EfficientNet with other deep learning models, consistently finding that EfficientNet outperforms its counterparts [11].

MobilenetV2, InceptionResNetV2, ResNet-101, Xception, DenseNet and NasNetMobile were some of the convolutional neural network models used by Izadi et al. in their investigation to identify the presence of fungus. In their study, Mehra et al. employed deep learning models including DenseNet121, ResNet50, VGG-19 and VGG-16 to categorize and diagnose two nail disorders. In order to categorize fingernail disorders, Maniyan et al. collected 13 characteristics from nail colour, shape, and texture and used the KNN classifier.

Yani et al. conducted a study using the Inception-V3 architecture to investigate the presence of Terry's Nail. Four different types of nail diseases, including nail hyperpigmentation, healthy nails, nail fungus and nail clubbing were looked at in another research by Abdulhadi (2021). The classification of the samples was done using a dataset and five pretrained deep convolutional neural network (CNN) models, namely ResNet50, AlexNet, GoogleNet, DenseNet201 and Vgg16. Using ResNet50, the greatest recognition rate of 96.40% was attained. The emphasis of the conventional strategy of Safira et al. was on locating anomalies in Terry's Nail. The grey level co-occurrence matrix (GLCM) and the KNN classifier were used to analyze the textural aspects. With K=1, the best accuracy result was 70.93%.

3 Methodology

The primary focus is on leveraging Convolutional Neural Networks (CNNs) to automatically extract meaningful features from nail images and predict the presence of diseases. The proposed methodology demonstrates promising results in terms of accuracy and efficiency, highlighting the potential for non-invasive and early disease detection. VGG16 (Visual Geometry Group 16) is a deep convolutional neural network architecture that was developed for image classification tasks. It gained popularity due to its simplicity and effectiveness in image recognition tasks. It consists of 16 convolutional and fully connected layers, with a fixed architecture. Following figure illustrates the overview of proposed method.

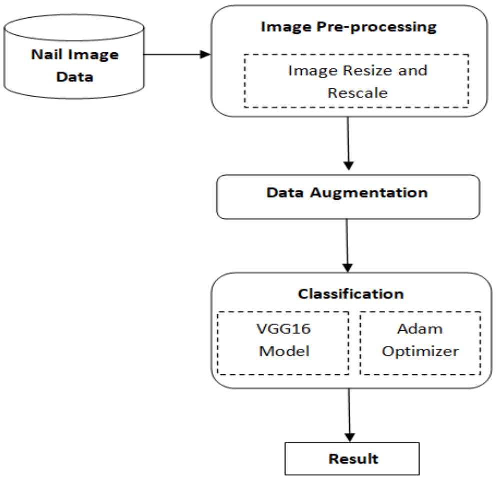
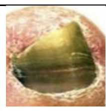
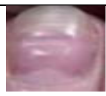
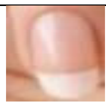


Fig.1. Overview of proposed model

3.1 Data Collection

For the classification purpose, data collected through digital camera and categorize on the basis of different colours of nails displayed in table 1.

TABLE 1: Classes of Nail Database

Image Classes	Class name	No. of Images	Sample Image
1	Black	2612	
2	Blue	1449	
3	Copper	3234	
4	Grey	543	
5	Green	1154	
6	Purple	356	
7	Red	1395	
8	White	3778	
9	Yellow	3504	
10	Healthy	711	
Total No. of images		18736	

3.2Image Pre-processing

Resizing all images of dataset to 224x224 is an essential preprocessing step when using the VGG16 model for image classification. VGG16 was designed with a fixed input size of 224x224 pixels, and resizing ensures that all input images conform to this requirement. During the resizing process, images are adjusted to the specified dimensions while preserving their aspect ratio. This standardized input size is essential because it enables the model to effectively learn and extract features from images, as the convolutional layers are optimized to work with this specific input shape. Resizing guarantees consistent data format and aids in achieving accurate and reliable predictions from theVGG16 model.

3.3Data Augmentation

VGG16 is designed to learn hierarchical features from images, but they require a substantial amount of diverse data to generalize well to various scenarios. Data augmentation artificially increases the dataset size by applying transformations to the existing images, effectively creating new training examples. This process exposes the model to different viewpoints, scales, and lighting conditions, making it more resilient to variations in real-world data and reducing overfitting, ultimately enhancing the model's performance and robustness. Data augmentation involves applying random transformations like rotations, flips, and shifts to training images. The dataset is divided into 80% and 20% for training and testing sets to train and evaluate the VGG16 model's performance respectively.

4 Modeling and Evaluation

4.1Convolutional Neural Network

Convolutional Neural Network (CNN) functions as an algorithm for both feature extraction and classification. It derives from the Multilayer Perceptron and mirrors the functioning of the human neural network. CNN's inception dates back to 1988 when YannLecun introduced it [22], marking a milestone in the emergence of Deep Learning. Specifically applied to image classification, CNN processes input images, subsequently classifying them into specific categories like airplanes, ships, birds, cats, and cows. Various CNN architectures, such as VGGNet, Alexnet, DenseNet, and ResNet, have since been developed. The core constituents of a CNN encompass the input layer, convolutional layer, activation layer, pooling layer, fully connected layer, and output layer [23] [24]. The convolutional layer extracts features, pooling layer generates new filters, and the fully connected layer in the output stage holds numerous neurons acting as decision-makers. Employing the CNN transfer learning model [23], a general strategy in computer vision, involves building systems by leveraging previously learned models for large-scale data challenges. This study harnesses transfer learning VGG16 models, pretrained on ImageNet, as a means to address data collection hurdles. The testing setup assesses the robustness of the proposed model's performance.

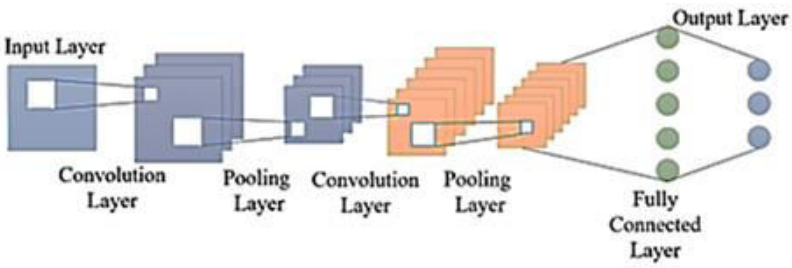


Fig.2. Fundamental Structure of CNN

4.2Visual Geometry Group-16(VGG-16)

VGG16 represents a widely employed Convolutional Neural Network (CNN) design within the realm of image recognition and classification [25]. Research has demonstrated that VGG16 can achieve notable accuracy levels in specific image classification scenarios [26], [27]. Simonyan and Zisserman introduced VGG16 in 2014

[28], and true to its name, this architecture comprises 16 layers [29]. Conceived as advancement over the AlexNet design, VGG16 innovatively employed smaller 3x3 kernel filters.

The fundamental structure of VGG16 is depicted in Figure 2. Simultaneously, Figure 3 illustrates the model parameters employed in this study, while Table 2 outlines the sequential processes undertaken by each layer within the VGG-16 model. During the training phase, a 224x224 RGB image undergoes two-dimensional convolution, utilizing 3x3 kernel dimensions. This convolutional process is enacted across successive layers, culminating in the extraction of a feature vector that significantly influences decision-making during the classification phase.

Fig.3. Fundamental Structure of VGG-16

5 Performance Evaluation

The assessment of the suggested approaches' effectiveness involves a comparison of the various pre trained models employed. This comparison is conducted through the computation of distinct metrics [17]. For the purpose of comparison, five metrics are computed. The initial metric is sensitivity (recall), which gauges the accuracy of labeling positive class instances. Mathematically, this can be expressed as follows:

$$\text{Sensitivity (Recall)} = TP/\ (TP + FN) \qquad (1)$$

Here, TP denotes true positives, signifying the count of correctly identified positive instances, while FN represents false negatives, indicating the quantity of positive cases misclassified. Specificity pertains to the likelihood of true negatives within a given alternate class condition. Essentially, it estimates the probability of a true negative label. This concept is denoted by:

$$\text{Specificity} = TN/\ (TN + FP) \qquad (2)$$

Where TN represents the count of true negatives, indicating negative cases that are correctly identified as negative and FP corresponds to false positives, characterizing instances of negative class that are erroneously classified as positive. Broadly discourse, sensitivity and specificity scale the algorithm's performance with respect to individual classes; sensitivity pertains to the positive class, while specificity pertains to the negative class. Frequently, accuracy stands out as the prevailing metric for assessing classification performance [18]. During the evaluation phase, accuracy was computed at intervals of 20 iterations. This measurement determines the proportion of samples that are accurately classified and can be symbolized as:

$$\text{Accuracy} = (TP + TN)/\ (TP + TN + FP + FN) \qquad (3)$$

Conversely, precision is defined as the quotient of true positives divided by the sum of true positives and false positives. This concept is articulated as:

$$\text{Precision} = TP/\ (TP + FP) \qquad (4)$$

This metric concerns accuracy; it assesses the forecasting capability of the algorithm. Precision gauges the degree of accuracy in the predicted positives, specifically how many of them are truly positive. The ultimate measurement is the F1-score, which calculates the balanced mean of precision and recall using their harmonic mean. Its definition is as follows:

F1-score = 2 × (Precision × Recall)/ (Precision + Recall)   (5)

It canters' its analysis on the positive class. A notable value of this metric signifies enhanced performance of the model in relation to the positive class.

TABLE 2: Step by Step execution of VGG-16

Layer	Step
1	Conv. 64 filters
2	Conv. 64 filters + Max pooling
3	Conv. 128 filters
4	Conv. 128 filters + Max pooling
5	Conv. 256 filters
6	Conv. 256 filters
7	Conv. 256 filters + Max pooling
8	Conv. 512 filters
9	Conv. 512 filters
10	Conv. 512 filters + Max pooling
11	Conv. 512 filters
12	Conv. 512 filters
13	Conv. 512 filters + Max pooling
14	Fully connected with 4096 nodes
15	Fully connected with 4096 nodes
16	Output layer with Softmax activation with 1000 nodes.

```
Model: "vgg16"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 224, 224, 3)]     0

 block1_conv1 (Conv2D)       (None, 224, 224, 64)      1792

 block1_conv2 (Conv2D)       (None, 224, 224, 64)      36928

 block1_pool (MaxPooling2D)  (None, 112, 112, 64)      0

 block2_conv1 (Conv2D)       (None, 112, 112, 128)     73856

 block2_conv2 (Conv2D)       (None, 112, 112, 128)     147584

 block2_pool (MaxPooling2D)  (None, 56, 56, 128)       0

 block3_conv1 (Conv2D)       (None, 56, 56, 256)       295168

 block3_conv2 (Conv2D)       (None, 56, 56, 256)       590080

 block3_conv3 (Conv2D)       (None, 56, 56, 256)       590080

 block3_pool (MaxPooling2D)  (None, 28, 28, 256)       0

 block4_conv1 (Conv2D)       (None, 28, 28, 512)       1180160

 block4_conv2 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_conv3 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_pool (MaxPooling2D)  (None, 14, 14, 512)       0

 block5_conv1 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv2 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv3 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_pool (MaxPooling2D)  (None, 7, 7, 512)         0

=================================================================
Total params: 14,714,688
Trainable params: 14,714,688
Non-trainable params: 0
_________________________________________________________________
```

Fig.4. Model Parameters

6 Result and Discussion

In this research, a dataset comprising a total of 18,736 images of nails was employed for both training and testing purposes. The testing phase involved utilizing transfer learning techniques with specific parameters, employing a batch size of 20 and experimenting with various epochs. Graphs illustrating the training progression of the suggested model, aimed at categorizing nail images into ten distinct classes, are presented in Figure 4and corresponding confusion matrix in figure 5, specifically focusing on the tenth epoch. The outcomes derived from the proposed model unveil the attainment of a peak accuracy rate of 94.27% when discerning among the ten different classes of nail images. These results, which are detailed in Table 3, underscore the efficacy of the model.

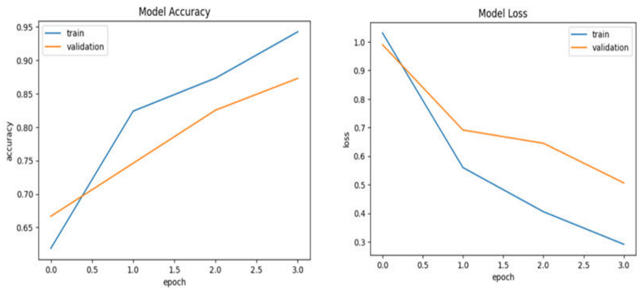


Fig.4.Human disease classification performance during the training process

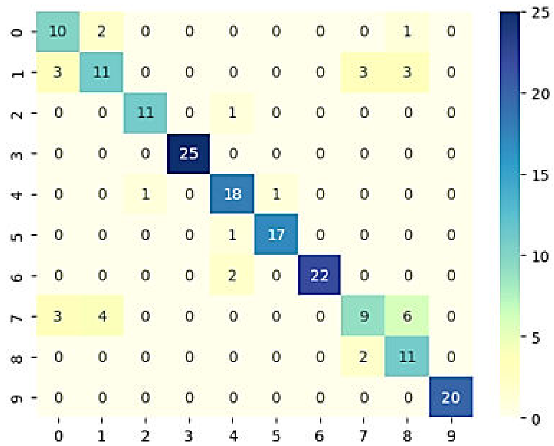


Fig.5. Confusion Matrix of classification process

TABLE 3: The testing results of Human disease classification into 10 classes using VGG-16

Image Class	precision	recall	f1-score
0	0.62	0.77	0.69
1	0.65	0.55	0.59
2	0.92	0.92	0.92
3	1.00	1.00	1.00
4	0.82	0.90	0.86
5	0.94	0.94	0.94
6	1.00	0.92	0.96
7	0.64	0.41	0.50
8	0.52	0.85	0.65
9	1.00	1.00	1.00

Furthermore, a comparative analysis was conducted, encompassing a comparison with prior investigations as detailed in Table 4. The method introduced in this study for the classification of nail diseases exhibited superior performance, particularly in the scenario involving three categories of classification, surpassing the outcomes of earlier research efforts.

TABLE 4: Comparison with previous studies

Study	Method	Classification classes	Accuracy (%)
Abdulhadi et. al [19]	Alexnet	4	92.5%
Indi and Gunge [20]	RGB Analysis	5	65%
Nijhawan et. al [21]	hybrid of Convolutional Neural Network	11	84.58%
This Study	VGG16	3	94.27%

7 Conclusion

Study attains a remarkable 94.27% accuracy using the VGG16 model across ten distinct image classes. This achievement underscores the model's efficacy in complex image classification tasks. The outcomes hold significant potential for medical applications, particularly in the realm of human disease diagnosis. The model's superior performance compared to alternatives underscores its value in enhancing diagnostic precision and patient care. While this study marks a significant advancement, further investigations could explore its generalizability and performance in diverse clinical scenarios. Ultimately, this research contributes a valuable tool for refining medical diagnostics through advanced image classification, promising improved healthcare practices.

Reference

1. *Lv, Qing, Suzhen Zhang, and Yuechun Wang.(2022) "Deep Learning Model of Image Classification Using Machine Learning." Advances in Multimedia 2022*
2. *Xin, M., Wang, Y. (2019) Research on image classification model based on deep convolution neural network. J Image Video Proc. 2019, 40*
3. *Wang, Xuesong, Chen Chen, Yuhu Cheng, and Z. Jane Wang (2016) "Zero-shot image classification based on deep feature extraction." IEEE Transactions on Cognitive and Developmental Systems 10, no. 2: 432-444.*
4. *Sánchez, Jorge, and FlorentPerronnin(2011) "High-dimensional signature compression for large-scale image classification." In CVPR 2011, pp. 1665-1672. IEEE.*
5. *Al-Saffar, Ahmed Ali Mohammed, Hai Tao, and Mohammed Ahmed Talab (2017) "Review of deep convolution neural network in image classification." In 2017 International conference on radar, antenna, microwave, electronics, and telecommunications (ICRAMET), pp. 26-31. IEEE.*
6. *Kim, Sang-Hyeon, and Han-Lim Choi (2019)"Convolutional neural network-based multi-target detection and recognition method for unmanned airborne surveillance systems." International Journal of Aeronautical and Space Sciences 20: 1038-1046.*
7. *Canny, J. A Computational Approach to Edge Detection. IEEE Trans. Pattern Anal. Mach. Intell. 1986, PAMI-8, 679–698.*
8. *Huang, Y.; Li, L. (2011) Naive Bayes Classification Algorithm Based on Small Sample Set. In Proceedings of the 2011 IEEE International Conference on Cloud Computing and Intelligence Systems, Beijing, China, 15–17 September; pp. 34–39.*
9. *Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. (1998) Gradient-Based Learning Applied to Document Recognition. Proc. IEEE 1998, 86, 2278–2324.*
10. *He, K.; Zhang, X.; Ren, S.; Sun, J. (2016) Deep Residual Learning for Image Recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.*
11. *Chan, T. F., Sandberg, B. Y., and Vese, L. A., (2000) Active contours without edges for vector- valued images. J. Vis. Commun. Image Represent. 11(2):130–141.*
12. *Egol, K. A., Koval, K. J., and Zuckerman, J. D., Handbook of fractures. Philadelphia: Lippincott Williams & Wilkins, 2010.*
13. *Fornaro, J., Szekely, G., and Harders, M., Semi-automatic segmentation of fractured pelvic bones for surgical planning.In: International symposium on biomedical simulation, pp. 82–89: Springer, 2010.*

14. *Gangwar, T., Calder, J., Takahashi, T., Bechtold, J. E., and Schillinger, D., Robust variational segmentation of 3D bone CT data with thin cartilage interfaces. Med. Image Anal. 47:95–110, 2018.*
15. *G. Naveen, Rai, A. Hussain, A. Kumar and R. Nijhawan, A leukonychia nail disease detection using deep learning, Int. J. Res. Appl. Sci. Eng. Technol., vol.8, no.3, pp.942-948, 2020.*
16. *L. Han, X. Zeng and L. Song, A novel transfer learning based on ALBERT for malicious network traffic classification, International Journal of Innovative Computing, Information and Control, vol.16, no.6, pp.2103-2119, 2020.*
17. *J. Abdulhadi, A. Al-Dujaili, A. J. Humaidi, and M. A. R. Fadhel, "Human nail diseases classification based on transfer learning," ICIC Express Letters, vol. 15, no. 12, pp. 1271–1282, 2021,*
18. *T. S. Indi and Y. A. Gunge, "Early Stage Disease Diagnosis System Using Human Nail Image Processing," International Journal of Information Technology and Computer Science, vol. 8, no. 7, pp. 30–35, 2016,*
19. *R. Nijhawan, R. Verma, Ayushi, S. Bhushan, R. Dua, and A. Mittal, "An integrated deep learning framework approach for nail disease identification," in Proceedings - 13th International Conference on Signal-Image Technology and Internet-Based Systems, SITIS 2017, 2018, vol. 2018-Janua, pp. 197–202,*
20. *Q. Zhang, M. Zhang, T. Chen, Z. Sun, Y. Ma, and B. Yu, "Recent advances in convolutional neural network acceleration," Neurocomputing, vol. 323, pp. 37–51, 2019,*
21. *M. S. Abou El-Seoud, M. H. Siala, and G. McKee, "Detection and Classifica-tion of White Blood Cells Through Deep Learning Techniques," International Journal of online and Biomedical Engineering, vol. 16, no. 15, pp. 94–105, 2020,*
22. *L. Alzubaidi et al., Review of deep learning: concepts, CNN architectures, challenges, applications, future directions, vol. 8, no. 1. Springer International Publishing, 2021.*
23. *D. Bendarkar, P. Somase, P. Rebari, R. Paturkar, and A. Khan, "Web Based Recognition and Translation of American Sign Language with CNN and RNN," International Journal of online and Biomedical Engineering, vol. 17, no. 1, pp. 34–50, 2021,*
24. *A. Elnakib, H. M. Amer, and F. E. Z. Abou-Chadi, "Early lung cancer detection using deep learning optimization," International Journal of online and Biomedical Engineering, vol. 16, no. 6, pp. 82–94, 2020,*
25. *Y. T. Li and J. I. Guo, "A VGG-16 based Faster RCNN Model for PCB Error Inspection in Industrial AOI Applications," in IEEE International Conference on Consumer Electronics-Taiwan, 2018, pp. 1–2.*
26. *K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, 2015, pp. 1–14.*
27. *B. R. Nanditha, A. G. Kiran, H. S. Chandrashekar, M. S. Dinesh, and S. Murali, "An Ensemble Deep Neural Network Approach for Oral Cancer Screening," International Journal of online and Biomedical Engineering, vol. 17, no. 2, pp. 121–134, 2021,*