

A Proposed Approach for Monkeypox Classification

Luong Hoang Huong, Nguyen Hoang Khang, Le Nhat Quynh, Le Huu Thang,
Dang Minh Canh, Ha Phuoc Sang
Department of Information Technology, FPT University, Can Tho, Viet Nam

Abstract—Public health concerns have been heightened by the emergence and spread of monkeypox, a viral disease that affects both humans and animals. The significance of early detection and diagnosis of monkeypox cannot be overstated, as it plays a crucial role in minimizing the negative impact on affected individuals and safeguarding public health. Monkeypox poses a considerable threat to human well-being, causing physical discomfort and mental distress, while also posing challenges to work productivity. This study proposes an applied model that combines deep learning models such as: ResNet-50, VGG16, MobileNet and machine learning models such as: Random Forest Classifier, K-Nearest Neighbors Classifier, Gaussian Naive Bayes Classifier, Decision Tree Classifier, Logistic Regression Classifier, AdaBoost Classifier to classify and detect monkeypox. The datasets are used in this research are the Monkeypox Skin Lesion Dataset (MSLD) and the Monkeypox Image Dataset (MID) that have total 659. Subjects range from healthy cases to severe skin lesions. The test results show that the model which combines deep learning and machine learning models achieves positive results, with Accuracy being 0.97 and F1-score being 0.98.

Keywords—Monkeypox; machine learning; deep learning; skin lesions

I. INTRODUCTION

Monkeypox is a viral zoonosis that can cause illness in humans and animals. The first recorded case of monkeypox in humans was in 1970 in the Democratic Republic of the Congo [1]. Since then, there have been several outbreaks of monkeypox, but the current outbreak is the largest and most widespread ever recorded [2]. The current outbreak of monkeypox has raised public health concerns due to its rapid spread and the potential for severe illness. Early detection and diagnosis of monkeypox is essential for minimizing the negative impact on affected individuals and safeguarding public health [3].

According to the available data [3], there have been 2891 confirmed monkeypox cases in the United States as of July 22, 2022. Globally, there have been a total of 71,237 laboratory-confirmed cases and 26 related deaths reported up to October 6th, 2022 [2]. Among the six World Health Organization regions, the Americas demonstrated the highest total laboratory-confirmed monkeypox cases (45,342 cases), followed by the European Region (24,889 cases), the African Region (727 cases), the Western Pacific Region (189 cases), the Eastern Mediterranean Region (67 cases), and the South-East Asia Region (23 cases) [2]. The nation with the highest cumulative monkeypox cases was the United States of America (26,723 cases), followed by Brazil (8,147 cases), Spain (7,209

cases), France (4,043 cases), the United Kingdom (3,654 cases), and Germany (3,640 cases) [2].

Traditional methods for detecting and diagnosing monkeypox, such as PCR testing, can be time-consuming and expensive. However, recent advancements in deep learning and machine learning models have provided an opportunity to develop an applied model for the classification and detection of monkeypox. By leveraging these technologies, the author aims to improve the efficiency and accuracy of monkeypox detection.

By combining the insights gained from the analysis of the dataset, the author can enhance the author's understanding of the current monkeypox outbreak and develop more effective strategies for its control and prevention. The proposed applied model, which integrates deep learning and machine learning models, holds promise for the timely and accurate classification and detection of monkeypox, thereby aiding in the mitigation of its impact on public health.

The test results show that the model which combines deep learning and machine learning models achieves positive results, with accuracy being 0.97 and F1-score being 0.98.

This research article has used the combination of deep learning models such as: ResNet-50, VGG16, MobileNet and machine learning models such as: Random Forest Classifier, K-Nearest Neighbors Classifier, Gaussian Naive Bayes Classifier, Decision Tree Classifier, Logistic Regression Classifier, AdaBoost Classifier to classify and detect monkeypox, with the foremost contributions of this paper are as follows:

- The author introduces an innovative and advanced solution to effectively address the challenge of detecting monkeypox.
- By combining deep learning and machine learning models, a formidable approach is devised to deliver exceptionally precise classification outcomes for monkeypox.
- Finding the best set of hyperparameters with fine-tuning.
- Remarkable headway is made in accurately classifying instances of the monkeypox disease, marking a significant stride forward in this area of research.

The article consists of five parts. Firstly, Section I serves as an introduction, providing a definition of the problem. Next, Section II presents related works that have been conducted. Moving on to Section III, the implementation method is

thoroughly explained. Section IV showcases the experimental results obtained from the research. Finally, in Section V, concluding remarks are provided to wrap up the article.

II. RELATED WORK

In this research [4], the author used these methods for data collection: Web-scraping for Image Collection, Expert Screening, Data Preprocessing, Augmentation. And using seven deep learning models named: ResNet-50, DenseNet121, Inception-V3, SqueezeNet, MnasNet-A1, MobileNet-V2, and ShuffleNet-V2-1x to conduct training on a dataset of diseases: Monkeypox, Chickenpox, Smallpox, Cowpox, Measles, Healthy; produced the best result in terms of accuracy (83%).

In addition, in [5], the author aims to compare different pre-trained deep learning (DL) models for Monkeypox virus detection on the disease dataset: Monkeypox, Chickenpox, Measles, Normal. Those deep learning models are VGG16, ResNet, Inception-V3, InceptionResNet, Xception, MobileNet, DenseNet, EfficientNet; produced the best result in average Precision, Recall, F1-score, and Accuracy of 85.44%, 85.47%, 85.40%, and 87.13% respectively.

Furthermore, in [6] researcher examines various deep convolutional neural network (CNN) models and several machine learning classifiers to detect monkeypox disease by analyzing skin images. Specifically, the study utilizes bottleneck features from three CNN models (AlexNet, GoogleNet, and Vgg16Net) in conjunction with multiple machines learning classifiers, including SVM, KNN, Naïve Bayes, Decision Tree, and Random Forest. The findings indicate that when using Vgg16Net features, the Naïve Bayes classifier achieves the highest accuracy rate of 91.11%.

Moreover, within [7], the author employs deep learning techniques to identify Monkeypox in digital images of skin. Various models including Support Vector Machines, ResNet-50, VGG16, SqueezeNet, and InceptionV3 were utilized. The skin data was acquired from Google using web-scraping techniques with Python's BeautifulSoup, SERP API, and requests libraries. The most successful model among them was VGG16, with an accuracy of 0.96 and an F1-score of 0.92.

In [8], the researcher employed several techniques to gather data, including web scraping, expert screening, data preprocessing, and data augmentation. Two deep learning models, namely AlexNet and VGG16, were utilized to train on a dataset comprising various diseases such as Monkeypox, Chickenpox, Measles, and Healthy. Notably, the VGG16 model yielded the highest accuracy, reaching 80%.

Additionally, in [9], the author the Kaggle Monkeypox Skin Lesion Dataset (MSLD) and the Monkeypox Skin Image Dataset (MSID) for their research purposes. Four deep neural networks were utilized for transfer learning, specifically Inception V3, ResNet 50 V2, MobileNet V2, and EfficientNet-B4. The MobileNet achieved superior performance compared to the other networks on the MSID dataset, achieving a balanced accuracy of 96.55%. Conversely, for the MSLD dataset, Inception V3 exhibited the most favorable metrics, achieving a balanced accuracy of 94%.

Within [10], the author aimed to integrate a well-trained deep learning (DL) algorithm and compare its performance against various other deep learning models. The disease dataset utilized in this research comprised Monkeypox and Chickenpox. The proposed convolutional neural network (CNN) model surpassed all other DL models, achieving a remarkable test accuracy of 99%. Moreover, a weighted average precision, recall, and F1 score of 99% were documented. Impressively, AlexNet demonstrated superior performance compared to other pre-trained models, achieving an accuracy of 98%. On the other hand, VGGNet, specifically VGG16 and VGG19, exhibited the lowest performance, with an accuracy of 80.00%. The ResNet-50 model attained an accuracy of 82%, while the InceptionV3 model achieved an accuracy of 89%.

Besides, in [11], the author assessed the effectiveness of five commonly used pre-trained deep learning models, namely VGG19, VGG16, ResNet-50, MobileNetV2, and EfficientNetB3. The experimental findings indicate that the MobileNetV2 model outperformed the others in terms of classification performance. It achieved an accuracy rate of 98.16%, a recall score of 0.96, a precision score of 0.99, and an F1-score of 0.98. Furthermore, when validating the model using different datasets, the MobileNetV2 model exhibited the highest accuracy of 94%.

And recently, within [12], the author introduced and assessed a revised version of the DenseNet-201 deep learning-based CNN model called MonkeyNet. By utilizing both the original and expanded datasets, this study put forward a deep convolutional neural network that demonstrated accurate identification of monkeypox disease. The accuracy achieved was 93.19% with the original dataset and 98.91% with the augmented dataset. The author employed the "MSID" dataset, which stands for "Monkeypox Skin Images Dataset."

Finally, in [13], the author utilized several custom models including MobileNetV3-s, EfficientNetV2, ResNet-50, VGG19, DenseNet121, and Xception models. Among these models, the hybrid MobileNetV3-s model, which was optimized, performed the most outstandingly. It achieved an average F1-score of 0.98, an AUC of 0.99, an accuracy of 0.96, and a recall of 0.97.

III. PROPOSED METHOD

A. Background

1) *ResNet-50*: ResNet stands for Residual Network and represents a distinctive variant of a convolutional neural network (CNN) that was first presented in the research paper titled "Deep Residual Learning for Image Recognition" in 2015. The authors of the paper, He Kaiming, Zhang Xiangyu, Ren Shaoqing, and Sun Jian, introduced this concept. CNNs are widely utilized in various computer vision applications. ResNet-50, on the other hand, is a specific instance of a convolutional neural network that consists of 50 layers, including 48 convolutional layers, one MaxPool layer, and one average pool layer. Residual neural networks are a type of artificial neural network (ANN) that constructs networks by

assembling residual blocks [14]. The deep residual learning framework of ResNet is shown in Fig. 1.

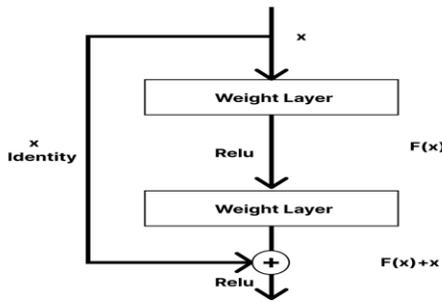


Fig. 1. Residual learning: a building block [15].

According to Fig. 1, the residual module in ResNet incorporates an identity mapping, resulting in a change of the network's output from $F(x)$ to $F(x) + x$. Typically, deep networks exhibit higher training errors compared to shallow networks. However, by appending multiple layers of constant mapping ($y = x$) to a shallow network, it can be transformed into a deep network, thereby achieving the same training error as the original shallow network. This indicates that layers with constant mapping are effectively trained. In the case of the residual network, when the residual is 0, the stacking layer essentially performs constant mapping. Based on the conclusion, it can be inferred that, at the very least, the network performance will not deteriorate theoretically [16].

2) *VGG16*: VGG16 refers to the VGG model, additionally referred to as VGGNet. It is a convolution neural network (CNN) model supporting 16 layers. K. Simonyan and A. Zisserman from Oxford University proposed this model and published it in a paper called "Very Deep Convolutional Networks for Large-Scale Image Recognition" [17]. VGG16 has been widely recognized as one of the top models in the ILSVRC-2014 competition, showcasing its superior performance. With the utilization of a dataset called ImageNet, which comprises over 14 million training images distributed across 1000 object classes, the VGG16 model achieves an impressive test accuracy of 92.7%. Notably, VGG16 builds upon the advancements made by AlexNet and introduces a significant improvement. Instead of employing larger filters, VGG16 replaces them with a series of smaller 3×3 filters. In comparison, AlexNet employs an 11-sized kernel for the initial convolutional layer and a 5-sized kernel for the second layer [18]. The architecture of VGG16 is shown in Fig. 2.

3) *MobileNet*: MobileNet is a CNN architecture designed for real-world applications, known for its efficiency and portability. Unlike previous architectures, MobileNets employ depthwise separable convolutions instead of standard convolutions to create lighter models. Additionally, MobileNets introduce two new global hyperparameters, namely width multiplier and resolution multiplier. These hyperparameters enable developers to make trade-offs between latency or accuracy, speed, and size according to

their specific needs [20][21]. The architecture of MobileNet is shown in Fig. 3.

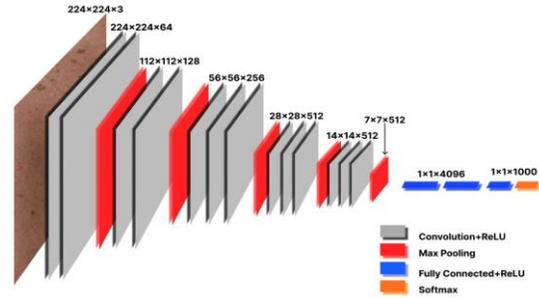


Fig. 2. VGG16 architecture [19].

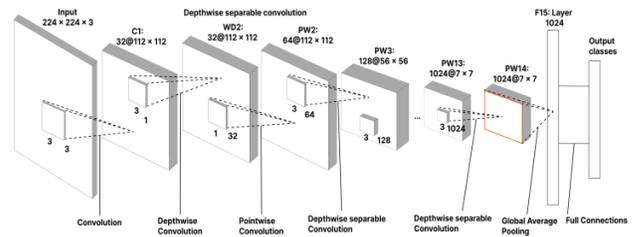


Fig. 3. MobileNet architecture.

4) *Random forest classification*: The Random Forest is a well-known supervised learning technique in machine learning. It is widely used for both Classification and Regression tasks. It operates on the principle of ensemble learning, which involves combining multiple classifiers to tackle complex problems and enhance the model's performance. The term "Random Forest" refers to a classification algorithm that comprises numerous decision trees constructed on different subsets of the provided dataset. By averaging the results from these trees, it aims to improve the predictive accuracy of the dataset. Instead of relying solely on a single decision tree, the random forest considers the predictions from each tree and determines the final output based on the majority votes among the predictions. Increasing the number of trees in the random forest enhances its accuracy and helps avoid overfitting issues [22]. The architecture of Random Forest Classification is shown in Fig. 4.

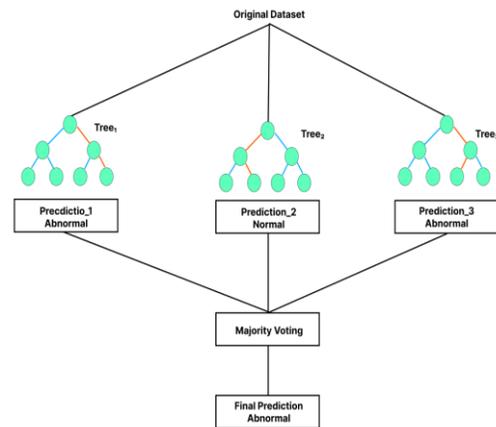


Fig. 4. Random forest classification architecture [23].

5) *K-nearest neighbors*: The K-Nearest Neighbors algorithm, referred to as KNN or k-NN, is a supervised learning classifier that operates on the principle of proximity. It is a non-parametric method commonly employed for classification tasks, where it determines the grouping of a given data point by comparing its proximity to other data points. Although it can handle both regression and classification problems, it is primarily used as a classification algorithm, based on the underlying idea that similar points tend to be located close to each other. In classification problems, a majority vote is used to assign a class label to a data point. This means that the label which appears most frequently around the given data point is chosen. Although this type of voting is technically known as "plurality voting," it is more commonly referred to as "majority vote" in literature. The difference between these terms lies in the fact that "majority voting" technically implies a majority greater than 50%, which is typically suitable when there are only two categories. [24]. The architecture of K-Nearest Neighbors is shown in Fig. 5.

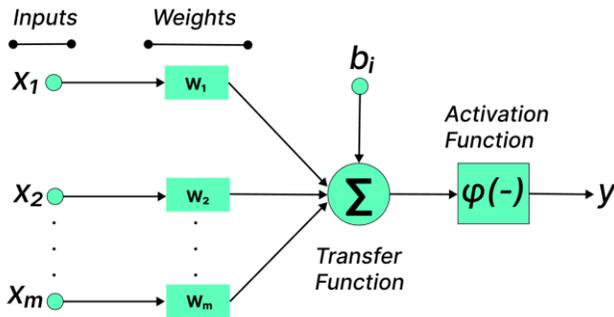


Fig. 5. K-nearest neighbors architecture [25].

6) *Gaussian Naive Bayes*: Gaussian Naive Bayes (GNB) is a machine learning classification method that relies on the probabilistic approach and Gaussian distribution. It operates under the assumption that each parameter (referred to as features or predictors) possesses an independent ability to predict the output variable. By combining the predictions from all parameters, the method produces a final prediction, which represents the probability of the dependent variable being classified into different groups. The group with the highest probability is assigned as the final classification. Gaussian Naive Bayes classifiers are a set of classification algorithms in supervised machine learning that rely on the principles of the Bayes theorem. They are a straightforward classification approach with notable effectiveness. They are particularly useful when dealing with datasets containing numerous input dimensions. Additionally, Naive Bayes classifiers can handle intricate classification tasks with success [26]. The Bayes Theorem is shown here.

The Formula for Bayes's Theorem Is

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(A) \cdot P(B|A)}{P(B)}$$

where:

$P(A)$ = The probability of A occurring

$P(B)$ = The probability of B occurring

$P(A|B)$ = The probability of A given B

$P(B|A)$ = The probability of B given A

$P(A \cap B)$ = The probability of both A and B occurring

7) *Decision tree classification*: A Decision Tree is a supervised learning technique used for classification and regression problems. It is a tree-like structure where internal nodes represent dataset features, branches are decision rules, and leaf nodes depict outcomes. Decision nodes make decisions with branches, while leaf nodes provide final outputs [27]. The general structure of a decision tree is shown in Fig. 6.

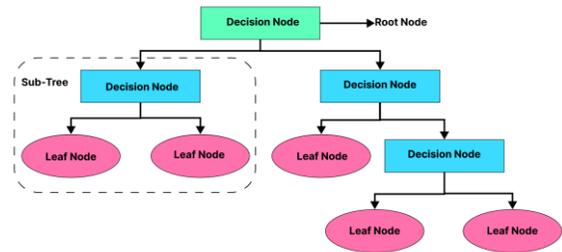


Fig. 6. Decision tree structure [27].

8) *Logistic regression*: Logistic Regression is an algorithm in Machine Learning used for classification purposes. It predicts the likelihood of specific classes by considering dependent variables. Essentially, the logistic regression model adds up the input features (often including a bias term) and applies the logistic function to the result. The output of logistic regression always falls between 0 and 1, making it suitable for binary classification tasks. A higher value indicates a greater probability of the current sample being classified as class=1, and vice versa [28]. The architecture of Logistic Regression is shown in Fig. 7.

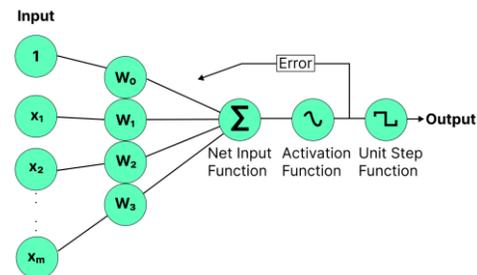


Fig. 7. Logistic regression architecture [29].

9) *AdaBoost classifier*: AdaBoost, also known as Adaptive Boosting, was introduced by Yoav Freund and Robert Schapire in 1996 as a type of ensemble boosting classifier. Its primary objective is to enhance the accuracy of classifiers by leveraging a combination of multiple classifiers. AdaBoost functions as an iterative ensemble technique, constructing a robust classifier by merge several classifiers that exhibit weak performance, thereby yielding a highly accurate and strong classifier. The fundamental idea underlying AdaBoost involves assigning weights to classifiers and training data samples during each iteration in a manner that guarantees precise predictions for atypical observations [30]. Fig. 8 illustrates the architecture of the lightweight CNN, which serves as the weak classifier in conjunction with AdaBoost.

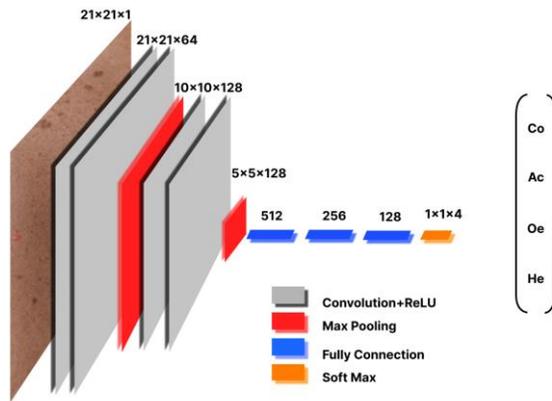


Fig. 8. AdaBoost [31].

B. Implementation Process

The implementation and model-building process of this research can be broken down into six main steps, which are outlined in Fig. 9.

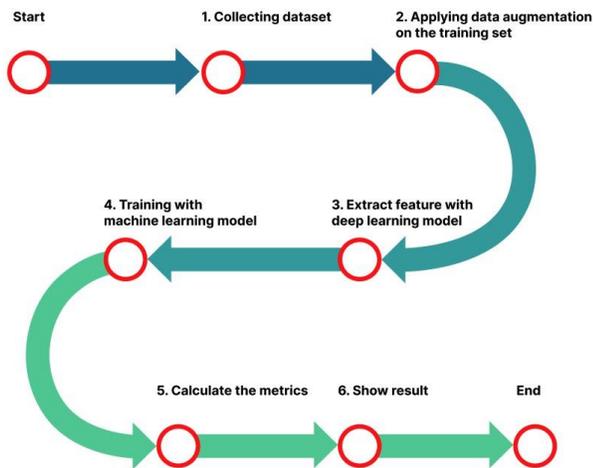


Fig. 9. Implementation process.

The initial step in this implementation involves the collection of data. The data was gathered by Sachin Kumar, Nafisa, Joydip Paul, Tazuddin Ahmed, and Tasnim Jahan Peana, and it was published in [32][33]. This dataset consists of

659 images in JPG format. The second step entails applying a data augmentation technique to the original dataset. This process aims to increase the dataset's size, thereby contributing to improved accuracy. Moving on to the next step, various deep learning models, such as ResNet-50, VGG16, and MobileNet, are employed to extract features from the dataset. These extracted features are then combined with machine learning models, including Random Forest Classification, K-Nearest Neighbors, Gaussian Naive Bayes, Decision Tree Classification, Logistic Regression, and AdaBoost Classifier, to produce highly accurate results. Once the training process is complete, the subsequent stage involves calculating and evaluating the accuracy of the models. Lastly, the final step focuses on presenting the achieved results.

C. The Proposed Architecture for Monkeypox Classification

In this research, the author proposes a combination model of deep learning model and machine learning model to obtain high-accuracy results. Initially, the author will gather monkeypox datasets from credible sources like Kaggle. To expand the dataset, the author will employ data augmentation techniques, resulting in a total of 4902 images. The subsequent phase involves extracting features from the dataset using deep learning models such as ResNet-50, VGG16, and MobileNet. After extracting the features, the author will merge them with various machine learning models, namely Random Forest Classification, K-Nearest Neighbors, Gaussian Naive Bayes, Decision Tree Classification, Logistic Regression, and AdaBoost Classifier. The next step involves obtaining the data from the training process performed by the machine learning models. The author will then calculate and evaluate this data to provide the final classification results for monkeypox. Through the author's proposed model, the author has obtained highly positive outcomes. Specifically, combining the MobileNet deep learning model with Logistic Regression machine learning model yielded the following results: precision of 0.99, recall of 0.98, F1-score of 0.98, and accuracy of 0.97. Fig. 10 illustrates the architecture the author proposes.

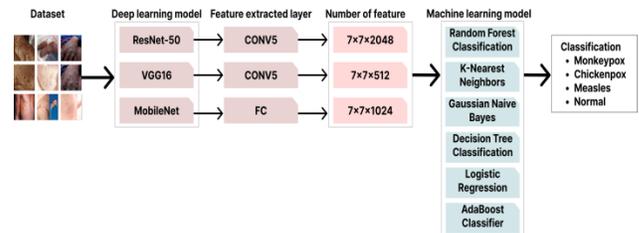


Fig. 10. The proposed architecture for monkeypox classification.

IV. EXPERIMENTS

In this section, the authors will provide an overview of the dataset utilized in the experiments, including its characteristics. Subsequently, the dataset will be employed in three separate experiments. The first experiment details the utilization of deep learning models for classifying four classes. The second experiment describes the author's proposed approach, which combines deep learning models with machine learning models to classify and detect Monkeypox. Lastly, the results obtained from the proposed model will be compared with state-of-the-art methods.

A. Dataset

In this section, the author conduct training on both deep learning models and machine learning models using two datasets: the Monkeypox Skin Lesion Dataset (MSLD) [32] and the Monkeypox Images Dataset (MID) [33]. These datasets have been subjected to data augmentation techniques, and the author compare the outcomes. The augmented dataset contains a larger amount of data compared to the original dataset. Specifically, the augmented dataset comprises 4902 images, while the original dataset consists of 659 images representing monkeypox, chickenpox, measles, and normal cases. The characteristics of the datasets are presented in Table I, and Fig. 11, respectively. The range of image dimensions is 256x256 pixels. Fig. 11 depicts the dataset's categories.



Fig. 11. Dataset illustration for three categories of monkeypox images.

TABLE I. DATASET CHARACTERISTICS

Labels	Images
Monkeypox	264
Chickenpox	100
Measles	80
Normal	215

B. Methods of Assessment and Comparison

This experiment involves comparing the performance of the author’s proposed architecture, which combines a deep learning model and a machine learning model, with well-known convolutional neural networks like ResNet-50, VGG16, and MobileNet, as well as state-of-the-art methods in terms of ACC and F1-score.

The study conducted three experiments. In Experiment 1, only deep learning models were used for training, evaluation, and testing. In Experiment 2, the author’s proposed model was employed for training, evaluation, and testing. The final experiment involved comparing the results of the proposed model with those obtained in Experiment 1 and with state-of-the-art methods.

C. Scenario 1: using Deep Learning Models to Classify Four Classes (Monkeypox, Chickenpox, Measles, Normal)

In this experiment, consistent hyperparameters were employed across all models, with epochs = {20}, batch sizes = {64}, and identical hidden layers for each model. The training outcomes for scenario 1 are presented in Table II.

TABLE II. RESULT OF USING DEEP LEARNING MODELS

DL Model	Pre	Recall	F1	ACC
ResNet50	0.78	0.72	0.69	0.72
MobileNet	0.97	0.97	0.97	0.97
VGG16	0.56	0.5	0.45	0.5

Based on the outcomes of the conducted experiment, it was observed that the MobileNet model achieved the most impressive performance, achieving accuracy of 0.97, F1-score of 0.97, recall of 0.97 and precision of 0.97. Conversely, the VGG16 model exhibited the poorest performance, achieving accuracy of 0.5, F1-score of 0.45, recall of 0.5, precision of 0.56. The confusion matrices for these two models represent the experiment's top and bottom results, as shown in Fig. 12 and 13, respectively.

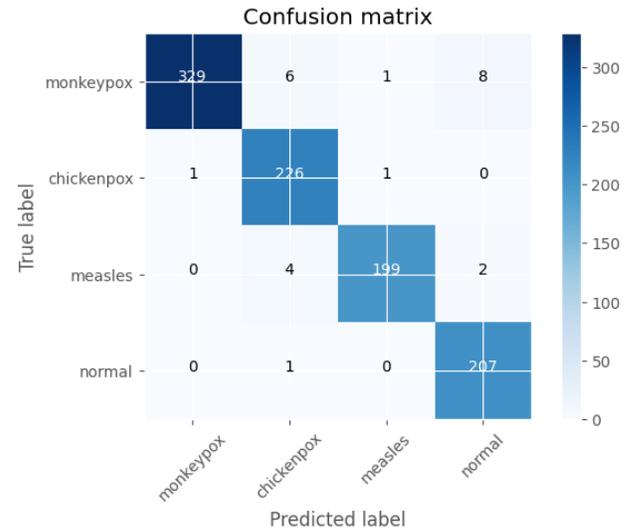


Fig. 12. The confusion matrix of MobileNet model.

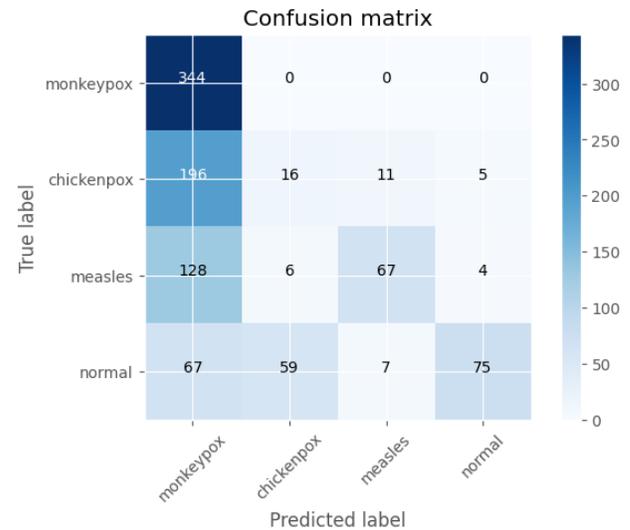


Fig. 13. The confusion matrix of VGG16 model.

D. Scenario 2: Using Deep Learning Models with Machine Learning Models to Classify Four Classes (Monkeypox, Chickenpox, Measles, Normal)

In this experiment, the author proposes a model by integrating deep learning and machine learning models. Specifically, deep learning models are used to extract features, which are then combined with machine learning models. The training outcomes for scenario 2 are presented in Table III.

TABLE III. RESULT OF USING DEEP LEARNING MODELS WITH MACHINE LEARNING MODELS

DL Model	ML Model	Dataset	Pre	Recall	F1	ACC
ResNet50	Random Forest Classification	Monkeypox	0.74	0.88	0.8	0.69
		Chickenpox	0.63	0.43	0.51	
		Measles	0.84	0.43	0.57	
		Normal	0.61	0.88	0.72	
	K-Nearest Neighbors	Monkeypox	0.8	0.85	0.82	0.73
		Chickenpox	0.59	0.6	0.6	
		Measles	0.78	0.66	0.71	
		Normal	0.71	0.75	0.73	
	Gaussian Naive Bayes	Monkeypox	0.73	0.27	0.39	0.39
		Chickenpox	0.27	0.84	0.41	
		Measles	0.5	0.09	0.15	
		Normal	0.62	0.42	0.5	
	Decision Tree Classification	Monkeypox	0.68	0.71	0.69	0.54
		Chickenpox	0.35	0.4	0.37	
		Measles	0.45	0.43	0.44	
		Normal	0.6	0.52	0.56	
	Logistic Regression	Monkeypox	0.85	0.85	0.85	0.73
		Chickenpox	0.59	0.57	0.58	
		Measles	0.7	0.69	0.7	
		Normal	0.71	0.74	0.73	
AdaBoost Classifier	Monkeypox	0.7	0.77	0.73	0.63	
	Chickenpox	0.45	0.45	0.45		
	Measles	0.66	0.37	0.47		
	Normal	0.66	0.81	0.72		
MobileNet	Random Forest Classification	Monkeypox	0.88	0.99	0.93	0.84
		Chickenpox	0.87	0.66	0.75	
		Measles	0.92	0.72	0.81	
		Normal	0.75	0.93	0.83	
	K-Nearest Neighbors	Monkeypox	0.46	0.9	0.61	0.57
		Chickenpox	0.76	0.51	0.61	
		Measles	0.98	0.49	0.65	
		Normal	0.61	0.22	0.32	
	Gaussian Naive Bayes	Monkeypox	0.9	0.92	0.91	0.79
		Chickenpox	0.63	0.71	0.67	
		Measles	0.83	0.66	0.73	
		Normal	0.79	0.81	0.8	
	Decision Tree Classification	Monkeypox	0.78	0.81	0.79	0.64
		Chickenpox	0.47	0.52	0.5	

DL Model	ML Model	Dataset	Pre	Recall	F1	ACC	
ResNet50		Measles	0.55	0.48	0.51		
		Normal	0.69	0.65	0.67		
	Logistic Regression	Monkeypox	0.99	0.98	0.98	0.97	
		Chickenpox	0.94	0.97	0.95		
		Measles	0.96	0.96	0.96		
		Normal	0.98	0.97	0.98		
	AdaBoost Classifier	Monkeypox	0.69	0.83	0.75	0.65	
		Chickenpox	0.47	0.45	0.46		
		Measles	0.6	0.51	0.55		
		Normal	0.79	0.7	0.75		
	VGG16	Random Forest Classification	Monkeypox	0.88	0.96	0.92	0.83
			Chickenpox	0.85	0.57	0.68	
Measles			0.9	0.74	0.81		
Normal			0.71	0.94	0.81		
K-Nearest Neighbors		Monkeypox	0.88	0.87	0.87	0.75	
		Chickenpox	0.81	0.6	0.69		
		Measles	0.53	0.94	0.68		
		Normal	0.86	0.51	0.64		
Gaussian Naive Bayes		Monkeypox	0.78	0.77	0.77	0.64	
		Chickenpox	0.51	0.37	0.43		
		Measles	0.49	0.59	0.54		
		Normal	0.63	0.7	0.67		
Decision Tree Classification		Monkeypox	0.77	0.77	0.77	0.66	
		Chickenpox	0.5	0.59	0.54		
		Measles	0.65	0.54	0.59		
		Normal	0.64	0.63	0.63		
Logistic Regression	Monkeypox	0.97	0.97	0.97	0.93		
	Chickenpox	0.9	0.84	0.87			
	Measles	0.92	0.95	0.94			
	Normal	0.89	0.92	0.9			
AdaBoost Classifier	Monkeypox	0.73	0.79	0.76	0.64		
	Chickenpox	0.47	0.48	0.47			
	Measles	0.54	0.41	0.47			
	Normal	0.68	0.72	0.7			

Based on the outcomes of the conducted experiment, it was observed that the combination of MobileNet model and Logistic Regression model achieved the most impressive performance, achieving accuracy of 0.97, F1-score of 0.98, recall of 0.98, and precision of 0.99. Conversely, the combination of ResNet50 model and Gaussian Naive Bayes model exhibited the poorest performance, achieving accuracy of 0.39, F1-score of 0.39, recall of 0.27, precision of 0.73. The

confusion matrices for these two models represent the experiment's top and bottom results, as shown in Fig. 14 and 15, respectively.

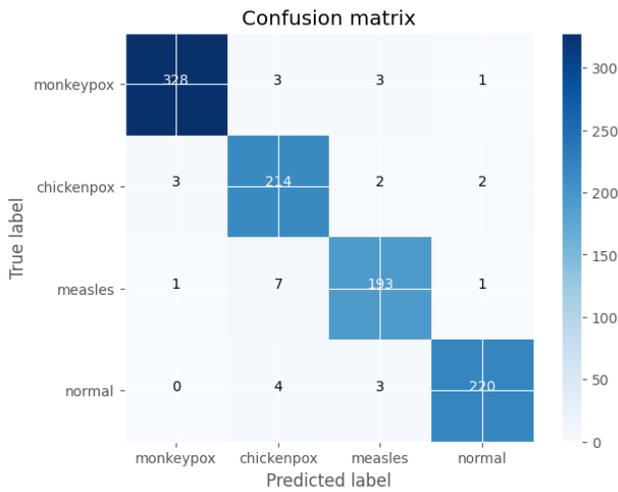


Fig. 14. The confusion matrix of combination between MobileNet and Logistic Regression model.

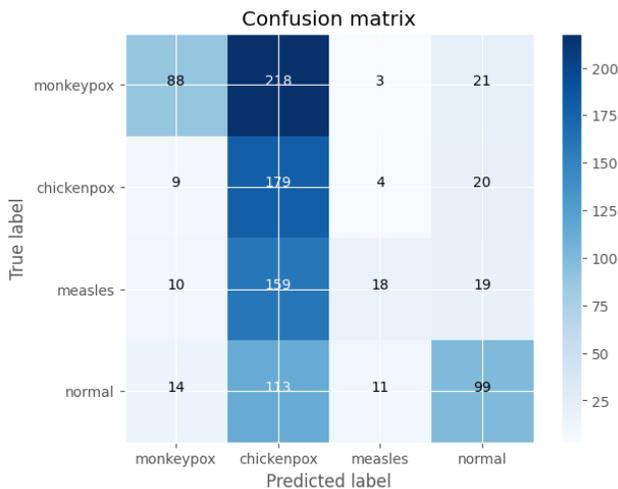


Fig. 15. The confusion matrix of combination between ResNet50 and Gaussian Naive Bayes model.

E. Scenario 3: Using Deep Learning Models to Classify and Detect Monkeypox

Table IV compares the performance of current methods for detecting Monkeypox using the author's proposed architecture. It is evident that the proposed model has achieved encouraging outcomes when compared to previous similar tasks. The above experimental results clearly indicate the suitability of the author's proposed model for image based Monkeypox classification. By combining the extracted features of the deep learning model and utilizing them in the machine learning model, the author's proposed approach outperforms other experimental models. Furthermore, the author conducts a comparison of the author's method with the average outcomes of ratios and state-of-the-art techniques. The confusion matrix of the proposed model in Scenario 2 and performance

comparison with previous similar studies as shown in Fig. 16 and Table IV.

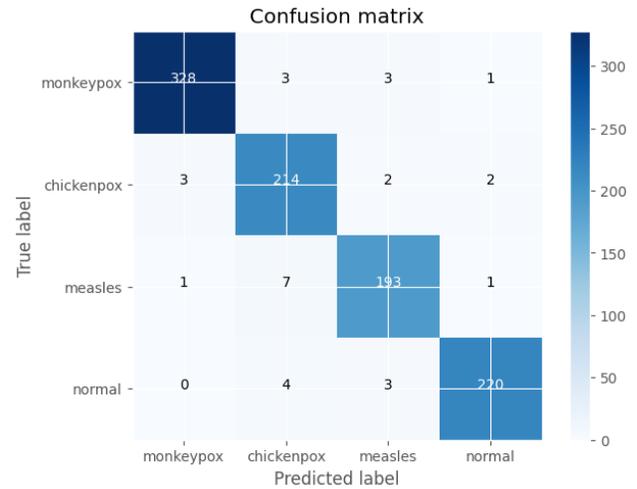


Fig. 16. The confusion matrix of the proposed model in Scenario 2 (combination between MobileNet and logistic regression model).

TABLE IV. PERFORMANCE COMPARISON WITH PREVIOUS SIMILAR STUDIES

Ref.	Dataset	Architecture	ACC
[4]	Web-scraping	CNN	0.83
[7]	Web-scraping	CNN	0.96
[8]	Web-scraping	CNN	0.8
[9]	MSLD, MSID	CNN (MobileNet V2)	0.96
[12]	MSID	CNN	0.93
Ours	MID, MSLD	MobileNet & Logistic Regression for 4 classes	0.97
Ours	MID, MSLD	MobileNet for 4 classes	0.97

V. CONCLUSION

Nowadays, health issues are becoming more and more important, especially concerns about sudden outbreaks of diseases and traditional disease detection methods which can be time-consuming and expensive. Therefore, it has highlighted people's need and interest in tools that support accurate and fast diagnosis. This research paper mentions the application of deep learning and machine learning models to support accurate and fast diagnosis of monkeypox. Through fine-tuning hyperparameters, this approach achieves remarkable accuracy in classification, signifying significant progress in monkeypox research. The author proposes to use a deep learning model called "A" to extract the feature from the dataset and then use that feature in conjunction with a machine learning model called "B" to get highly accurate results. For instance, the author use MobileNet deep learning model combined with Logistic Regression machine learning model gives very positive results: precision is 0.99, recall is 0.98, F1-score is 0.98 and accuracy is 0.97. These findings contribute to the development of rapid and accurate diagnostic tools, improving the detection and early diagnosis of monkeypox to minimize its negative impact on public health. Improved

diagnostics can also enhance patient outcomes and bolster global health security by strengthening the author's preparedness for infectious disease outbreaks. Collaboration among researchers, healthcare institutions, and governments is crucial to driving the widespread adoption of these advanced tools and creating a more resilient global health community. In the future, the author plans to incorporate the "segmentation" technique for image analysis, aiming to delineate the affected areas and enhance the accuracy of this proposed method. This addition will facilitate the precise identification of the diseased regions.

REFERENCES

- [1] "Mpx (monkeypox)." <https://www.who.int/news-room/fact-sheets/detail/monkeypox> (accessed Jul. 09, 2023).
- [2] S. Park and H. Yon, "Global, regional, and national incidence and mortality of human monkeypox infection in 107 countries and territories, October 2022: a systematic analysis for World Health Organization database and rapid review," *Life Cycle*, vol. 2, 2022, doi: 10.54724/lc.2022.e16.
- [3] D. Philpott, "Epidemiologic and Clinical Characteristics of Monkeypox Cases — United States, May 17–July 22, 2022," *MMWR Morb. Mortal. Wkly. Rep.*, vol. 71, 2022, doi: 10.15585/mmwr.mm7132e3.
- [4] T. Islam, M. A. Hussain, F. U. H. Chowdhury, and B. M. R. Islam, "Can Artificial Intelligence Detect Monkeypox from Digital Skin Images?" *bioRxiv*, p. 2022.08.08.503193, Oct. 27, 2022. doi: 10.1101/2022.08.08.503193.
- [5] C. Sitaula and T. B. Shahi, "Monkeypox Virus Detection Using Pre-trained Deep Learning-based Approaches," *J. Med. Syst.*, vol. 46, no. 11, p. 78, Oct. 2022, doi: 10.1007/s10916-022-01868-2.
- [6] V. Kumar, "Analysis of CNN features with multiple machine learning classifiers in diagnosis of monkeypox from digital skin images." *medRxiv*, p. 2022.09.11.22278797, Nov. 28, 2022. doi: 10.1101/2022.09.11.22278797.
- [7] O. Alrusaini, "Deep Learning Models for the Detection of Monkeypox Skin Lesion on Digital Skin Images," *Int. J. Adv. Comput. Sci. Appl.*, vol. 14, p. 637, Jan. 2023, doi: 10.14569/IJACSA.2023.0140170.
- [8] A. B. Ural, "A Computer-Aided Feasibility Implementation to Detect Monkeypox from Digital Skin Images with Using Deep Artificial Intelligence Methods," *Trait. Signal*, vol. 40, no. 1, pp. 383–388, Feb. 2023, doi: 10.18280/ts.400139.
- [9] M. F. Almufareh, S. Tehsin, M. Humayun, and S. Kausar, "A Transfer Learning Approach for Clinical Detection Support of Monkeypox Skin Lesions," *Diagnostics*, vol. 13, no. 8, Art. no. 8, Jan. 2023, doi: 10.3390/diagnostics13081503.
- [10] D. Uzun Ozsahin, M. T. Mustapha, B. Uzun, B. Duwa, and I. Ozsahin, "Computer-Aided Detection and Classification of Monkeypox and Chickenpox Lesion in Human Subjects Using Deep Learning Framework," *Diagnostics*, vol. 13, no. 2, Art. no. 2, Jan. 2023, doi: 10.3390/diagnostics13020292.
- [11] A. S. Jaradat et al., "Automated Monkeypox Skin Lesion Detection Using Deep Learning and Transfer Learning Techniques," *Int. J. Environ. Res. Public. Health*, vol. 20, no. 5, Art. no. 5, Jan. 2023, doi: 10.3390/ijerph20054422.
- [12] D. Bala et al., "MonkeyNet: A robust deep convolutional neural network for monkeypox disease detection and classification," *Neural Netw.*, vol. 161, pp. 757–775, Apr. 2023, doi: 10.1016/j.neunet.2023.02.022.
- [13] M. Altun, H. Gürüler, O. Özkaraca, F. Khan, J. Khan, and Y. Lee, "Monkeypox Detection Using CNN with Transfer Learning," *Sensors*, vol. 23, no. 4, Art. no. 4, Jan. 2023, doi: 10.3390/s23041783.
- [14] "ResNet-50: The Basics and a Quick Tutorial," *Datagen*. <https://datagen.tech/guides/computer-vision/resnet-50/> (accessed Jul. 12, 2023).
- [15] "Figure 6: Residual learning: a building block. Image from original..." *ResearchGate*. https://www.researchgate.net/figure/Residual-learning-a-building-block-Image-from-original-ResNet-paper-15_fig4_343414434 (accessed Jul. 21, 2023).
- [16] X. Feng, X. Gao, and L. Luo, "A ResNet50-Based Method for Classifying Surface Defects in Hot-Rolled Strip Steel," *Mathematics*, vol. 9, no. 19, Art. no. 19, Jan. 2021, doi: 10.3390/math9192359.
- [17] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition." *arXiv*, Apr. 10, 2015. doi: 10.48550/arXiv.1409.1556.
- [18] "Understanding VGG16: Concepts, Architecture, and Performance," *Datagen*. <https://datagen.tech/guides/computer-vision/vgg16/> (accessed Jul. 12, 2023).
- [19] "Figure 1. The VGG16 deep learning architecture [18]," *ResearchGate*. https://www.researchgate.net/figure/The-VGG16-deep-learning-architecture-18_fig1_339256141 (accessed Jul. 21, 2023).
- [20] "MobileNet V1 Architecture," *OpenGenus IQ: Computing Expertise & Legacy*, Oct. 07, 2020. <https://iq.opengenus.org/mobilenet-v1-architecture/> (accessed Jul. 12, 2023).
- [21] A. G. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications." *arXiv*, Apr. 16, 2017. doi: 10.48550/arXiv.1704.04861.
- [22] "Machine Learning Random Forest Algorithm - Javatpoint," *www.javatpoint.com*. <https://www.javatpoint.com/machine-learning-random-forest-algorithm> (accessed Jul. 12, 2023).
- [23] "Structure diagram of Random Forest algorithm," *ResearchGate*. https://www.researchgate.net/figure/Structure-diagram-of-Random-Forest-algorithm_fig1_366786501 (accessed Jul. 21, 2023).
- [24] "What is the k-nearest neighbors algorithm? | IBM." <https://www.ibm.com/topics/knn> (accessed Jul. 12, 2023).
- [25] "Figure 2. Architecture of the ANN network k-Nearest Neighbor (k-NN)..." *ResearchGate*. https://www.researchgate.net/figure/Architecture-of-the-ANN-network-k-Nearest-Neighbor-k-NN-Classifer_fig1_325209169 (accessed Jul. 21, 2023).
- [26] <https://www.facebook.com/prateek.majumder.5>, "Gaussian Naive Bayes," *OpenGenus IQ: Computing Expertise & Legacy*, Feb. 23, 2020. <https://iq.opengenus.org/gaussian-naive-bayes/> (accessed Jul. 12, 2023).
- [27] "Decision Tree Algorithm in Machine Learning - Javatpoint," *www.javatpoint.com*. <https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm> (accessed Jul. 12, 2023).
- [28] <https://www.facebook.com/kdnuggets>, "How Does Logistic Regression Work?," *KDnuggets*. <https://www.kdnuggets.com/how-does-logistic-regression-work.html> (accessed Jul. 12, 2023).
- [29] "Fig. 3. Schematic diagram for logistic regression classification," *ResearchGate*. https://www.researchgate.net/figure/Schematic-diagram-for-logistic-regression-classification_fig2_333982722 (accessed Jul. 21, 2023).
- [30] "AdaBoost Classifier Algorithms using Python Sklearn Tutorial." <https://www.datacamp.com/tutorial/adaboost-classifier-python> (accessed Jul. 12, 2023).
- [31] "The architecture for the light CNN, used as weak classifier with Adaboost," *ResearchGate*. https://www.researchgate.net/figure/The-architecture-for-the-light-CNN-used-as-weak-classifier-with-Adaboost_fig3_361150461 (accessed Jul. 21, 2023).
- [32] "Monkeypox Skin Lesion Dataset." <https://www.kaggle.com/datasets/nafin59/monkeypox-skin-lesion-dataset> (accessed Jul. 09, 2023).
- [33] "Monkeypox Images Dataset." <https://www.kaggle.com/datasets/sachinkumar413/monkeypox-images-dataset> (accessed Jul. 09, 2023).