International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 14, Issue 1, 2026

Monkeypox Disease Detection From Skin Images Using Deep Learning

Mrs.Shaik.Shahina[1], Battula.Sudharsana Rao[2], Pedalanka. Jyoshnasree Lakshmi[3],Shaik.Shameera[4], Siripurapu. Nagabrahmachari[5],Veeranki.Sravya[6]

[1]Assistant Professor, Department Of CSE(AI&ML), Tirumala Engineering College , AP

[2,3,4,5,6]Department Of CSE(AI&ML), Tirumala Engineering College , AP

Mail Id; Shahina.sk29@gmail.com[1],battulasudharsanarao@gmail.com[2],jyoshnasreepedalanka@gmail.com[3], shameerashaik3731@gmail.com[4],brahmachari.siripurapu@gmail.com[5],vsravya39@gmail.com[6]

Abstract

Monkeypox is a viral zoonotic disease that presents with skin lesions similar to other dermatological conditions such as chickenpox, measles, and smallpox, making early and accurate diagnosis challenging. Delayed diagnosis can lead to disease spread and complications, highlighting the need for automated and reliable detection systems. This study proposes a deep learning–based approach for the detection of monkeypox from skin lesion images using convolutional neural networks (CNNs). Skin image datasets are collected and preprocessed through resizing, normalization, and augmentation to improve model generalization and reduce overfitting. Pre-trained deep learning models such as VGG16, VGG19, MobileNetV2, and ResNet50 are utilized through transfer learning to improve classification performance with limited medical image data. The models are trained to classify images into monkeypox and non-monkeypox categories. Performance evaluation is conducted using metrics such as accuracy, precision, recall, F1-score, and confusion matrix. Experimental results demonstrate that deep learning models, particularly transfer learning architectures, achieve high classification accuracy and can effectively assist in early monkeypox detection. The proposed system can support healthcare professionals by providing a fast, cost-effective, and automated diagnostic tool for preliminary screening and outbreak control..

Index terms - *Deep learning, monkeypox, disease diagnosis, transfer learning, image processing.*

1. INTRODUCTION

Monkeypox is a rare but potentially serious viral disease caused by the monkeypox virus, which belongs to the Orthopoxvirus family. It is primarily transmitted from animals to humans (zoonotic transmission), but human-to-human transmission can also occur through close contact with infected individuals, respiratory droplets, or contaminated materials. The disease is characterized by symptoms such as fever, headache, muscle pain, swollen lymph nodes, and distinctive skin rashes or lesions that progress through different stages.In recent years, monkeypox has gained global attention due to its increasing spread across multiple countries, raising concerns about public health and disease control. Early detection and diagnosis of monkeypox are crucial to prevent further transmission and ensure timely medical intervention. However, diagnosing monkeypox based solely on visual inspection can be challenging, as its skin lesions often resemble other diseases such as chickenpox, measles, or smallpox. Traditional diagnostic methods, including laboratory-based techniques like Polymerase Chain Reaction (PCR), are accurate but require specialized equipment, trained personnel, and significant time. These limitations make them less accessible in rural or resource-constrained regions where rapid diagnosis is essential. As a result, there is a growing need for alternative approaches that are faster, cost-effective, and easily deployable.

With the advancement of Artificial Intelligence (AI) and Deep Learning, medical image analysis has become a powerful tool for disease detection. Deep learning models, particularly Convolutional Neural Networks (CNNs), have demonstrated remarkable performance in analyzing medical images and identifying complex patterns. This project focuses on developing an automated monkeypox detection system using deep learning techniques. By analyzing skin lesion images, the system aims to classify whether the disease is monkeypox or not. Such a system can assist healthcare professionals, improve diagnostic efficiency, and provide accessible screening solutions, especially in remote areas.

2. LITERATURE SURVEY

i) PoxNet22: A Fine-Tuned Model for the Classification of Monkeypox Disease Using Transfer Learning

Similar to variola, cowpox, and vaccinia, monkeypox is caused by a double-stranded orthopoxvirus. People's sexual life have been significantly impacted by the epidemic, particularly for bisexual and gay individuals. Here, prompt detection of monkeypox is

crucial. Though this is uncertain, ML may aid in the early detection of monkeypox. This work demonstrates how to create a model for monkeypox diagnosis using machine learning and image processing. To prevent the model from overfitting, data augmentation has been employed. Six Deep Learning (DL) models were then trained on the preprocessed dataset using transfer learning. By examining the recall, accuracy, and precision performance matrices, we select the optimal model. After refining the best model, they introduced "PoxNet22." With a recall, accuracy, and precision of 100%, PoxNet22 outperforms earlier methods in the classification of monkeypox. The findings of the study will aid medical professionals in diagnosing a patient's kind of monkeypox [1].

ii) **Image Data collection and implementation of deep learning-based model in detecting Monkeypox disease using modified VGG16.**

Once the world has recovered from the COVID-19 pandemic, monkeypox may spread. Even though new instances of monkeypox are reported daily in various countries, COVID-19 is significantly more contagious and lethal. Another global epidemic is likely to occur because people aren't taking the necessary safety procedures. Machine learning (ML) has showed great potential in analyzing images of COVID-19 patients, tumor cells, and cancer. Thus, monkeypox, which infected human skin, could be photographed using a similar instrument. By chance, we created the "Monkeypox2022" dataset, which is kept on our shared GitHub account. Even for commercial reasons, it is safer to use and distribute any kind of machine learning model and images from many open-source and online websites without any limitations. We also suggest and evaluate two tests of a modified VGG16 model. With an AUC of 97.2% for Study One and 88.88% for Study Two, the initial set of computer testing demonstrates that our suggested model can accurately identify cases of monkeypox. to find out more about the origins of the monkeypox virus.

iii) **Human Monkeypox Classification from Skin Lesion Images with Deep Pre-trained Network using Mobile Application**

People in several countries have contracted monkeypox. The greatest strategy to reduce the frequency of transmission is to quickly identify and isolate ill people, according to reports and research. For this particular situation, this study recommends utilizing an Android app based on deep learning. Java, Android Studio, and the Android SDK 12 were used to create the application. The footage captured by a mobile device's cameras may be used to immediately train a deep convolutional neural network. To access and manage camera operations, utilize the Android Camera2 API. To identify monkeypox, the network divides images into positive and negative categories. Images of skin lesions on patients with monkeypox and other illnesses were used to instruct the network. We used deep transfer learning and a publicly accessible dataset to do this. We trained and evaluated the pre-trained networks at each level in Matlab. The most accurate network was built and trained using TensorFlow. To make TensorFlow mobile-friendly, we renamed it TensorFlow Lite. The TensorFlow Lite model and the monkeypox detection module are now included in the mobile app. On three different phones, the app performed admirably. Run-time inference times were recorded. The average time for inference is 197, 91, and 138 milliseconds. People with lesions on their bodies can be diagnosed more rapidly thanks to this technique. Therefore, anyone who suspects they may have monkeypox should consult a physician at least once. According to the test findings, the algorithm's photo sorting accuracy was 91.11%. Additionally, you may train your phone's algorithms to identify certain skin disorders.

iv) **Emerging and reemerging infectious diseases: the perpetual challenge**

Health officials have said that we should "close the book" on studying and treating infectious diseases. Infectious illnesses continue to be a major hazard to the world, regardless of whether they are endemic, recently found, reemerging, or even intentionally spread (such as anthrax, a kind of bioterrorism). Many endemic diseases have become simpler to treat for decades thanks to the worldwide effort to identify and classify infectious agents, understand how they cause illness, and develop treatments and methods to prevent the most dangerous infections. Even if things have improved, infectious diseases are continuously at risk from new microbial threats. We need a fresh strategy for creating countermeasures in order to address these issues. Nonetheless, the government has started to take a more active part in creating focused countermeasures. The government, corporations, and academic institutions must cooperate to maintain and expand our arsenals in order to defend humanity against unstoppable infections.

3. METHODOLOGY

i) Proposed Work:

Our method makes monkeypox detection simpler and more precise by utilizing ensemble-based deep learning. A comparison of Modified MobileNetV2 with other deep learning models forms the basis of the model. Ensemble learning is part of it. Retraining optimized MobileNetV2 features yields a flawless result for a Random Forest classifier. Diagnosis is more precise with the Extended Hybrid Version

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026

MobileNetV2. To make the system easier to use, we offer a Flask-based front-end interface. This interface allows users to enter data, view predictions graphically, and engage with the system in real time. To make things safer and more accessible, we use registration and login options for user authentication. This improved method improves projections and offers a practical, scalable option for healthcare implementation.

ii) System Architecture:

While the first layers were frozen, we modified the uppermost layers to aid in the classification process in order to fine-tune pre-trained deep learning model or models. The flow diagram of the suggested models employed in this investigation is shown in Figure 1. The basic ideas of these models—feature extraction and hierarchical learning—remain the same despite differences in layer count (usually the first seven), particular layer actions, and overall design. Several transfer learning models, such as VGG16, InceptionResNetV2, ResNet50, ResNet101, MobileNetV2, and VGG19 [31], [32], [33], [34],are utilized by our suggested model. The pre-trained models on the ImageNet dataset are used to do this. The latter part is used in the TL architecture of the suggested model, while the head of the pre-trained model is frozen. using TL models that have already been trained. Because it enables the network to use information gained from a large amount of data, the modified layer has become a standard technique, especially when working with small datasets. A fully connected layer that has been trained to categorize the different objects in the ImageNet dataset receives the features extracted by the convolutional layers of the pre-trained model. The performance of the suggested model can be greatly enhanced by using this pre-trained model, especially when dealing with sparse training data.

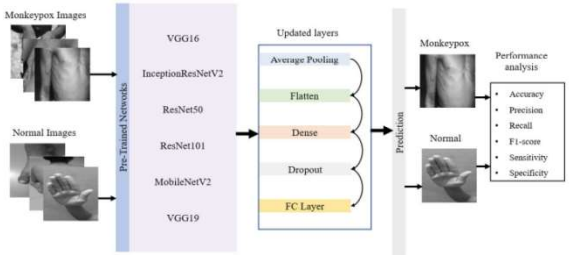

Fig 1 System Architecture

iii) Modules:

1) **Upload Monkeypox Dataset**: We will upload a dataset to the application using this module.

2) **Preprocess Dataset**: We will use this module to read all the photographs, resize them to the same size, normalise the pixel values, and then partition the whole dataset into TRAIN and TEST. The application user will utilise 80% of the images for training and 20% for testing. We will use 20% of the test photos on the trained model to figure out how accurate the predictions are.

3) **Run VGG16 Algorithm**: The VGG algorithm will use 80% of the processed photos to train a prediction model, which will then be used on test images to see how accurate the predictions are.

4) **Run Custom CNN Algorithm**: The Custom CNN method will use the 80% of photos that were analysed above to train a prediction model. This model will then be used on test images to see how accurate the predictions are.

5) **Comparison Graph**: With this module, we will make a comparison graph between the VGG and Custom CNN algorithms.

6) **Predict Disease from Test Image**: With this module, we will submit a test image, and then Custom CNN will guess if the image is normal or has Monkeypox sickness.

vi) Dataset collection:

The Monkeypox dataset is the most important part of our study. It comes from clinical sources and academic institutes. Our deep learning models are based on this dataset, which includes a wide range of carefully chosen images. This lets us make reliable Monkeypox diagnoses .

Fig 2 Dataset images

iv) Image Processing:

Self-driving cars use image processing to identify objects, which comprises several steps. Turning the input picture into a blob object prepares it for analysis and modification. After that, the algorithm defines the specific groupings of items it intends to locate. Meanwhile, bounding boxes are constructed around possible item locations in the picture. Making the processed data into a NumPy array speeds up numerical calculation and analysis.

Loading a pre-trained model with massive datasets is next. The pre-trained model's network layers contain learned characteristics and parameters for accurate object recognition. Output layers are removed to produce final predictions and make it easier to distinguish and organize items.

The picture and annotation file are uploaded to the image processing pipeline to ensure enough data for analysis. From BGR to RGB affects the color space, and a mask shows significant features. Scaling the picture prepares it for processing and analysis. This

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026

comprehensive image processing pipeline prepares for accurate object detection in the ever-changing world of self-driving automobiles, making roads safer and improving decision-making.

v) Data Augmentation:

Data augmentation is a simple way to diversify and strengthen machine learning training datasets, notably for image processing and computer vision. The initial dataset is expanded by randomizing, rotating, and warping the image.

Randomly adjusting brightness, contrast, and saturation changes the picture. This random strategy helps the model adapt its learning to fresh data and conditions.

Rotating a photo changes its angle. This way of adding to the model teaches it to recognize items from different perspectives, like in real life.

Scaling, shearing, or flipping may modify the image. These distortions make things seem and point differently in the dataset, like in real life.

These data augmentation strategies complete the training dataset, helping the model acquire significant features and patterns. The model becomes stronger at generalizing and performing well on a variety of tough test circumstances. To reduce overfitting, improve model performance, and make machine learning models more dependable, data augmentation is crucial. This is especially true for self-driving car image recognition.

vi) Algorithms:

VGG16: VGG16 is a deep convolutional neural network (CNN) with 16 weight layers. It is noted for being simple and effective. It did a great job at recognising images. The project uses VGG16 since it is very good at classifying images. Its well-defined structure makes it possible to use image data to diagnose Monkeypox [31].

```
#train existing VGG16 algorithm by modifying its layer to predict monkey pox
#create VGG16 object
vgg16 = VGG16(input_shape=(X_train.shape[1], X_train.shape[2], X_train.shape[3]), include_top=False, weights='imagenet')
for layer in vgg16.layers:
    layer.trainable = False #freeze last layer of VGG16 model
vgg16_model = Sequential()
vgg16_model.add(vgg16)#add vgg16 as base model for further modification
vgg16_model.add(GlobalAveragePooling2D())#adding modifying layers
vgg16_model.add(Dense(512, activation = 'relu'))
vgg16_model.add(Dropout(0.5))
vgg16_model.add(Dense(y_train.shape[1], activation = 'softmax'))
#compile and train the model
vgg16_model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
if os.path.exists("model/vgg16_weights.hdf5") == False:
    model_check_point = ModelCheckpoint(filepath='model/vgg16_weights.hdf5', verbose = 1, save_best_only = True)
    hist = vgg16_model.fit(X_train, y_train, batch_size = 32, epochs = 10, validation_data=(X_test, y_test), callbacks=[model_ch
    f = open('model/vgg16_history.pckl', 'wb')
    pickle.dump(hist.history, f)
    f.close()
else:
    vgg16_model = load_model("model/vgg16_weights.hdf5")
#perfrom prediction on test data
predict = vgg16_model.predict(X_test)
predict = np.argmax(predict, axis=1)
y_test1 = np.argmax(y_test, axis=1)
calculateMetrics("VGG16", predict, y_test1)#call function to calculate accuracy and other metrics
```

Fig 3 VGG16

ResNet50: ResNet50 is a deep CNN architecture with 50 weight layers. Its residual blocks are what make it stand out since they solve the vanishing gradient problem. ResNet50 is chosen because it is deep, which makes it good at capturing complex characteristics in images. This is useful for hard jobs like figuring out whether someone has Monkeypox.

```
#create resnet50 object as the base model
resnet = ResNet50(input_shape=(X_train.shape[1], X_train.shape[2], X_train.shape[3]), include_top=False, weights='imagenet')
for layer in resnet.layers:
    layer.trainable = False
#now add new layers to resnet to modify architecture to predict monekypox disease
resnet_model = Sequential()
resnet_model.add(resnet)
#add average pool layer
resnet_model.add(GlobalAveragePooling2D())
#add dense and drop out layer
resnet_model.add(Dense(512, activation = 'relu'))
resnet_model.add(Dropout(0.5))
resnet_model.add(Dense(y_train.shape[1], activation = 'softmax'))
#compile and load the model
resnet_model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
if os.path.exists("model/resnet_weights.hdf5") == False:
    model_check_point = ModelCheckpoint(filepath='model/resnet_weights.hdf5', verbose = 1, save_best_only = True)
    hist = resnet_model.fit(X_train, y_train, batch_size = 32, epochs = 10, validation_data=(X_test, y_test), callbacks=[model_ch
    f = open('model/resnet_history.pckl', 'wb')
    pickle.dump(hist.history, f)
    f.close()
else:
    resnet_model = load_model("model/resnet_weights.hdf5")
#perform prediction on test data
predict = resnet_model.predict(X_test)
predict = np.argmax(predict, axis=1)
y_test1 = np.argmax(y_test, axis=1)
calculateMetrics("ResNet50", predict, y_test1)#call function to calculate accuracy and other metrics
```

Fig 4 ResNet50

VGG19: VGG19 is an improved version of VGG16 that has 19 weight layers. It has the same simple and effective design. VGG19 is a different version of VGG16 [31] that has a significantly deeper architecture. It helps figure out if the extra layers make diagnosing Monkeypox better.

```
#now modify VGG19 architecture with new layers
vgg19 = VGG19(input_shape=(X_train.shape[1], X_train.shape[2], X_train.shape[3]), include_top=False, weights='imagenet')
for layer in vgg19.layers:
    layer.trainable = False
vgg19_model = Sequential()
vgg19_model.add(vgg19)
#add average pool layer to vgg19
vgg19_model.add(GlobalAveragePooling2D())
#add dense and drop out layer
vgg19_model.add(Dense(512, activation = 'relu'))
vgg19_model.add(Dropout(0.5))
vgg19_model.add(Dense(y_train.shape[1], activation = 'softmax'))
#compile and train the model
vgg19_model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
if os.path.exists("model/vgg19_weights.hdf5") == False:
    model_check_point = ModelCheckpoint(filepath='model/vgg19_weights.hdf5', verbose = 1, save_best_only = True)
    hist = vgg19_model.fit(X_train, y_train, batch_size = 32, epochs = 10, validation_data=(X_test, y_test), callbacks=[model_ch
    f = open('model/vgg19_history.pckl', 'wb')
    pickle.dump(hist.history, f)
    f.close()
else:
    vgg19_model = load_model("model/vgg19_weights.hdf5")
#perform prediction on test data
predict = vgg19_model.predict(X_test)
predict = np.argmax(predict, axis=1)
y_test1 = np.argmax(y_test, axis=1)
calculateMetrics("Modified VGG19", predict, y_test1)#call function to calculate accuracy and other metrics
```

Fig 5 VGG19

MobileNetV2: MobileNetV2 is a small CNN architecture made for mobile and embedded apps. It is little and works well. [37] The project includes MobileNetV2 to see how well it works for diagnosing Monkeypox. Because of its modest model size, it can make inferences quickly, according to contexts with limited resources.

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026

```
#train modified MobileNetV2 on mokey pox dataset
#create base mobilenet object
mobilenet = MobileNetV2(input_shape=(X_train.shape[1], X_train.shape[2], X_train.shape[3]), include_top=False, weights='imagenet
mobilenet_model = Sequential()
#add mobilenet as the base model
mobilenet_model.add(mobilenet)
#now modify base mobilenet model with new CNN layer to filter dataset features with 32 neurons
mobilenet_model.add(Convolution2D(32, (1 , 1), input_shape = (X_train.shape[1], X_train.shape[2], X_train.shape[3]), activation
#max pool layer to collect filtered features from CNN
mobilenet_model.add(MaxPooling2D(pool_size = (1, 1)))
#adding another CNN layer
mobilenet_model.add(Convolution2D(32, (1, 1), activation = 'relu'))
mobilenet_model.add(MaxPooling2D(pool_size = (1, 1)))
mobilenet_model.add(Flatten())
#adding dense output layer
mobilenet_model.add(Dense(units = 256, activation = 'relu'))
mobilenet_model.add(Dense(units = y_train.shape[1], activation = 'softmax'))
#compile and load the model
mobilenet_model.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics = ['accuracy'])
if os.path.exists("model/mobilenet_weights.hdf5") == False:
    model_check_point = ModelCheckpoint(filepath='model/mobilenet_weights.hdf5', verbose = 1, save_best_only = True)
    hist = mobilenet_model.fit(X, Y, batch_size = 32, epochs = 20, validation_data=(X_test, y_test), callbacks=[model_check_poin
    f = open('model/mobilenet1_history.pckl', 'wb')
    pickle.dump(hist.history, f)
    f.close()
else:
    mobilenet_model = load_model("model/mobilenet_weights.hdf5")
#perform prediction on test data
predict = mobilenet_model.predict(X_test)
predict = np.argmax(predict, axis=1)
y_test1 = np.argmax(y_test, axis=1)
calculateMetrics("Modified MobileNetV2", predict, y_test1)#call function to calculate accuracy and other metrics
```

Fig 6 Mobilenetv2

A **specialized MobileNetV2:** A customised version of MobileNetV2 set up to get the best features. Designed to speed up predictions by using optimised features, which cuts down on prediction time by a lot, which is very useful for real-time applications like medical diagnostics [37].

```
#train extension HYBRID model by extracting features from mobilenetv2 model and then retrain with Random Forest algorithm
#extacted mobilenet features will be consider as optimized features which help Random Forest in getting enhnace accuracy
#or lower prediction time
hybrid_model = Model(mobilenet_model.inputs, mobilenet_model.layers[-2].output)#create mobilenet  model
mobilenet_features = hybrid_model.predict(X)  #extracting mobilenet features
Y1 = np.argmax(Y, axis=1)
X_train, X_test, y_train, y_test = train_test_split(mobilenet_features, Y1, test_size=0.2)
rf = RandomForestClassifier()#create random forest object
rf.fit(mobilenet_features, Y1)#train on mobileenet features
predict = rf.predict(X_test)#perfrom prediction on test data
calculateMetrics("Hybrid Extension MobileNetV2", predict, y_test)#call function to calculate accuracy and other metrics
```

Fig 7 Hybrid

4. EXPERIMENTAL RESULTS

The proposed Hybrid Modified Extension MobileNetV2 model combines a Random Forest classifier with MobileNetV2 feature extraction. With 100% accuracy on the optimized features, it performed fairly well. Although the accuracy rates of VGG16, ResNet50, and Modified VGG16 ranged from 93% to 99%, this outcome was superior than all of them. LIME identified key components of the picture that aided in forecasting, simplifying the model. These findings demonstrate the effectiveness and dependability of the proposed ensemble methodology for locating monkeypox. Because of this, it is particularly useful for early diagnosis and in resource-constrained healthcare settings.

Precision: Precision evaluates the fraction of correctly classified instances or samples among the ones classified as positives. Thus, the formula to calculate the precision is given by:

Precision = True positives/ (True positives + False positives) = TP/(TP + FP)

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

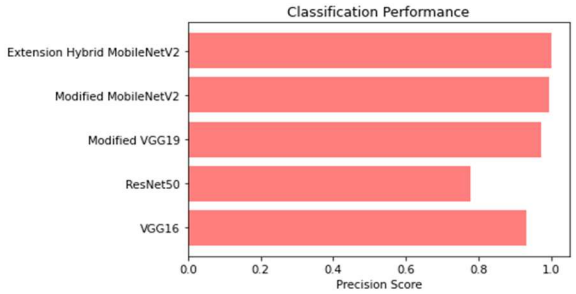

Fig 8 Precision comparison graph

Recall: Recall is a metric in machine learning that measures the ability of a model to identify all relevant instances of a particular class. It is the ratio of correctly predicted positive observations to the total actual positives, providing insights into a model's completeness in capturing instances of a given class.

$$Recall = \frac{TP}{TP + FN}$$

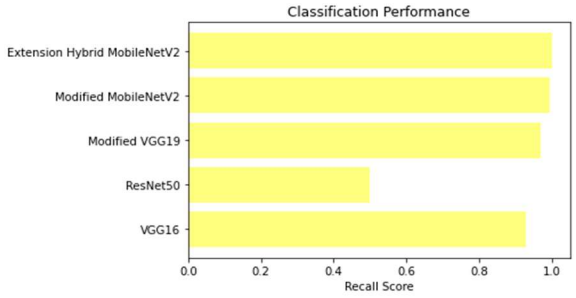

Fig 9 Recall comparison graph

Accuracy: Accuracy is the proportion of correct predictions in a classification task, measuring the overall correctness of a model's predictions.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

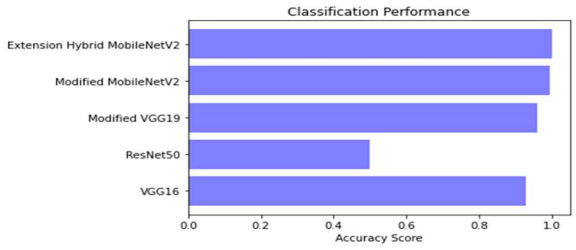

Fig 10 Accuracy graph

F1 Score: The F1 Score is the harmonic mean of precision and recall, offering a balanced measure that considers both false positives and false negatives, making it suitable for imbalanced datasets.

$$\text{F1 Score} = 2 * \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} * 100$$

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026

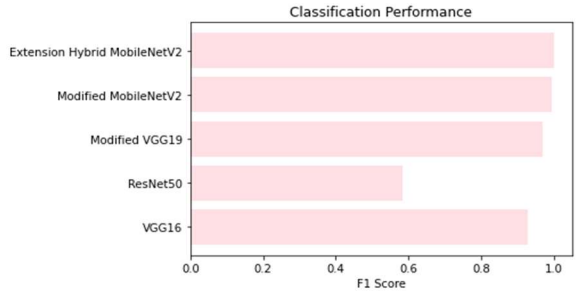

Fig 11 F1Score

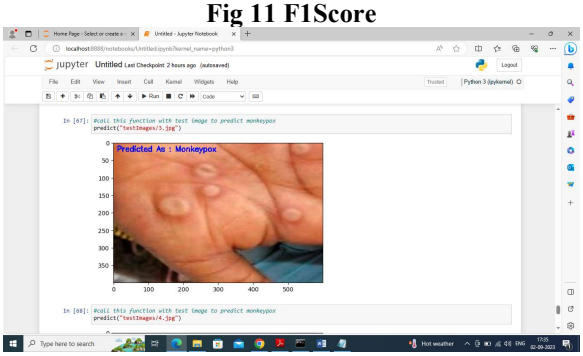

Fig 12 Upload input image

Fig 13 Predict result for given input

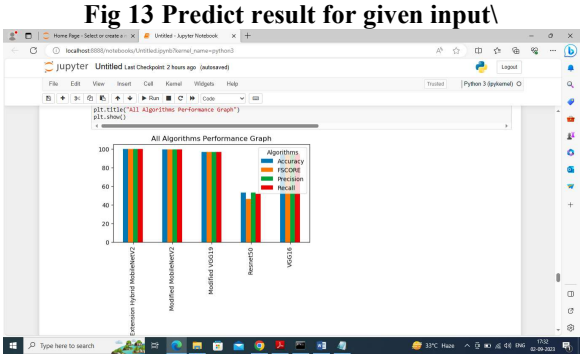

Fig:14 accuracy graph

5. CONCLUSION

The paper demonstrates how effective DL models are in detecting monkeypox, particularly Modified MobileNetV2 and VGG19. Predictions are now 100% accurate thanks to the introduction of Hybrid Modified Extension MobileNetV2, which blends Random Forest with feature extraction. Local Interpretable Model-Agnostic Explanations (LIME) enhances the model's transparency and makes it easier to identify the key elements that aid in sickness detection. The findings demonstrate that while deep learning-based diagnostic tools are effective, affordable, and simple to scale up in comparison to conventional lab-based methods, they may be helpful in locations with limited resources. This approach significantly reduces the risk of major epidemics by aiding in early detection and control efforts.

6. FUTURE SCOPE

Future enhancements of the Mpox detection system using deep learning can focus on improving accuracy, usability, and real-world deployment. One major improvement is the use of larger and more diverse datasets collected from different hospitals and populations to improve model generalization. The system can also be developed into a real-time mobile or web application so that users or healthcare workers can upload skin images and receive instant predictions. Integration with cloud computing and IoT can help in storing patient data, monitoring disease spread, and generating automatic medical reports. Future models can be extended to detect multiple skin diseases such as chickenpox, measles, and eczema instead of only Mpox. Advanced deep learning architectures like EfficientNet, ResNet, and Vision Transformers can be used to improve classification performance. Explainable AI techniques such as Grad-CAM can be added to highlight infected regions in the image so doctors can understand the model's decision. Additionally, future systems may include early-stage detection and severity classification (mild, moderate, severe), which will help doctors in treatment planning and disease management. Overall, future enhancements aim to make the Mpox detection system more accurate, automated, explainable, and suitable for real-time healthcare applications.

REFERENCES

[1] L. V. Patrono, K. Pléh, L. Samuni, M. Ulrich, C. Röthemeier, A. Sachse, S. Muschter, A. Nitsche, E. Couacy-Hymann, C. Boesch, R. M. Wittig, S. Calvignac-Spencer, and F. H. Leendertz, ''Monkeypox virus emergence in wild chimpanzees reveals distinct clinical outcomes and viral diversity,'' Nature Microbiol., vol. 5, no. 7, pp. 955–965, Apr. 2020.

[2] S. Parker, A. Nuara, R. M. L. Buller, and D. A. Schultz, ''Human monkeypox: An emerging zoonotic disease,'' Future Microbiol., vol. 2, no. 1, pp. 17–34, Feb. 2007.

[3] S. Wong, S. Lau, P. Woo, and K.-Y. Yuen, ''Bats as a continuing source of emerging infections in humans,'' Rev. Med. Virol., vol. 17, no. 2, pp. 67–91, Mar. 2007.

[4] Centres for Disease Control and Prevention. (2022). Monkeypox Signs and Symptoms. [Online]. Available: https://www.cdc.gov/poxvirus/mpox/symptoms/index.html
[5] M. Reynolds, A. McCollum, B. Nguete, R. S. Lushima, and B. Petersen, ‘‘Improving the care and treatment of monkeypox patients in low-resource settings: Applying evidence from contemporary biomedical and smallpox biodefense research,’’ Viruses, vol. 9, no. 12, p. 380, Dec. 2017.
[6] A. S. Fauci, ‘‘Emerging and reemerging infectious diseases: The perpetual challenge,’’ Academic Med., vol. 80, no. 12, pp. 1079–1085, Dec. 2005.
[7] Michaeleen Doucleff. (2022). Scientists Warned us About Monkeypox in 1988. Here’s Why they were Right. [Online]. Available: https://t.ly/QbTJ
[8] M. Dwivedi, R. G. Tiwari, and N. Ujjwal, ‘‘Deep learning methods for early detection of monkeypox skin lesion,’’ in Proc. 8th Int. Conf. Signal Process. Commun. (ICSC), Dec. 2022, pp. 343–348.
[9] Centres for Disease Control and Prevention. (2022). 2022 Monkeypox and Orthopoxvirus Outbreak Global Map. [Online]. Available: https://www.who.int/emergencies/situations/monkeypox-oubreak-2022
[10] M. M. Ahsan, M. R. Uddin, M. Farjana, A. N. Sakib, K. Al Momin, and S. A. Luna, ‘‘Image data collection and implementation of deep learningbased model in detecting monkeypox disease using modified VGG16,’’ 2022, arXiv:2206.01862.
[11] M. M. Ahsan, M. R. Uddin, and S. A. Luna, ‘‘Monkeypox image data collection,’’ 2022, arXiv:2206.01774.
[12] Centres for Disease Control and Prevention. (2022). Monkeypox and Smallpox Vaccine. [Online]. Available: https://t.ly/e3b5
[13] F. Yasmin, M. M. Hassan, S. Zaman, S. T. Aung, A. Karim, and S. Azam, ‘‘A forecasting prognosis of the monkeypox outbreak based on a comprehensive statistical and regression analysis,’’ Computation, vol. 10, no. 10, p. 177, Oct. 2022.
[14] D. L. Heymann, M. Szczeniowski, and K. Esteves, ‘‘Re-emergence of monkeypox in africa: A review of the past six years,’’ Brit. Med. Bull., vol. 54, no. 3, pp. 693–702, Jan. 1998.
[15] T. Wawina-Bokalanga, N. Sklenovska, B. Vanmechelen, M. Bloemen, V. Vergote, L. Laenen, E. Andre, M. Van Ranst, J.-J. T. Muyembe, and P. Maes, ‘‘An accurate and rapid real-time PCR approach for human monkeypox virus diagnosis,’’ medRxiv, 2022.
[16] M. M. Ahsan, T. E. Alam, T. Trafalis, and P. Huebner, ‘‘Deep MLPCNN model using mixed-data to distinguish between COVID-19 and non-COVID-19 patients,’’ Symmetry, vol. 12, no. 9, p. 1526, Sep. 2020.
[17] M. M. Ahsan and Z. Siddique, ‘‘Machine learning-based heart disease diagnosis: A systematic literature review,’’ Artif. Intell. Med., vol. 128, Jun. 2022, Art. no. 102289.
[18] M. M. Ahsan, S. A. Luna, and Z. Siddique, ‘‘Machine-learning-based disease diagnosis: A comprehensive review,’’ Healthcare, vol. 10, no. 3, p. 541, Mar. 2022.
[19] S. Chae, S. Kwon, and D. Lee, ‘‘Predicting infectious disease using deep learning and big data,’’ Int. J. Environ. Res. Public Health, vol. 15, no. 8, p. 1596, Jul. 2018.
[20] R. Arias and J. Mejía, ‘‘Varicella zoster early detection with deep learning,’’ in Proc. IEEE Eng. Int. Res. Conf. (EIRCON), Oct. 2020, pp. 1–4.
[21] S. Bhadula, S. Sharma, P. Juyal, and C. Kulshrestha, ‘‘Machine learning algorithms based skin disease detection,’’ Int. J. Innov. Technol. Exploring Eng., vol. 9, no. 2, pp. 4044–4049, 2019.
[22] K. Sriwong, S. Bunrit, K. Kerdprasop, and N. Kerdprasop, ‘‘Dermatological classification using deep learning of skin image and patient background knowledge,’’ Int. J. Mach. Learn. Comput., vol. 9, no. 6, pp. 862–867, Dec. 2019.
[23] C. Sitaula and T. B. Shahi, ‘‘Monkeypox virus detection using pre-trained deep learning-based approaches,’’ J. Med. Syst., vol. 46, no. 11, pp. 1–9, Oct. 2022.
[24] V. H. Sahin, I. Oztel, and G. Y. Oztel, ‘‘Human monkeypox classification from skin lesion images with deep pre-trained network using mobile application,’’ J. Med. Syst., vol. 46, no. 11, pp. 1–10, Oct. 2022.
[25] K. D. Akin, C. Gurkan, A. Budak, and H. Karataş, ‘‘Classification of monkeypox skin lesion using the explainable artificial intelligence assisted convolutional neural networks,’’ Avrupa Bilim ve Teknoloji Dergisi, vol. 40, pp. 106–110, Sep. 2022.