ISSN 2319-5991 www.ijerst.com

Peer Reviewed, Referred & Indexed Journal

Research Paper

A MobileNetV2-Based Hybrid Deep Learning Framework for High-Accuracy Monkeypox Detection and Real-Time Web Application

Ms.K.Baby Ramya[1], Mrs.K.Pavani[2], R.Gurunaidu[3]

#1 Assistant Professor in the Department of MCA, SRK Institute of Technology, Vijayawada.

#2 Assistant Professor & Head of Department of MCA, SRK Institute of Technology, Vijayawada

#3 Student in the Department of MCA, SRK Institute of Technology, Vijayawada

Abstract: The improved deep learning-based system for automated monkeypox detection utilizing transfer learning techniques is presented in this research. The categorization of skin lesion photos is assessed using many pre-trained convolutional neural network models, such as VGG16, ResNet50, VGG19, and MobileNetV2. A Hybrid MobileNetV2 model is suggested to enhance diagnostic performance. It achieves better accuracy than current models by combining enhanced feature extraction and classification capabilities. Additionally, a Flask-based web application connected with user authentication is added to the system to enable secure and real-time diagnostics. Experiments show that the suggested hybrid model performs exceptionally well, which makes it appropriate for quick, affordable, and easily accessible monkeypox detection, particularly in settings with limited resources.

Index terms - *Monkeypox Detection, Deep Learning, Transfer Learning, Hybrid MobileNetV2, Convolutional Neural Networks (CNN), Skin Lesion Classification, Flask Web Application, Medical Image Analysis, Automated Diagnosis, Explainable AI.*

1. INTRODUCTION

Early identification is difficult because the symptoms of monkeypox, an infectious disease caused by the Monkeypox virus (MPXV), resemble those of other skin-related disorders including chickenpox and smallpox. Serious public health concerns have been raised in recent years by the global spread of monkeypox, particularly in areas with poor medical infrastructure. Conventional diagnostic approaches, such as clinical examination and laboratory-based methods like PCR, are frequently costly, time-consuming, and equipment-intensive, which causes delays in diagnosis and treatment.

Automated illness detection using medical imaging has received a lot of interest with the development of artificial intelligence, especially deep learning. Convolutional Neural Networks (CNNs) have shown great efficacy in accurately classifying skin lesion photos by extracting complicated information. By using pre-trained models like VGG16, ResNet50, VGG19, and MobileNetV2, transfer learning further improves this process by cutting training time and boosting performance even with little datasets.

This study proposes a hybrid MobileNetV2-based deep transfer learning model to increase the precision and dependability of monkeypox detection. In order to offer a real-time, safe, and user-friendly diagnostic platform, a Flask-based web application with user authentication is also created. The goal of this comprehensive method is to provide a quick, affordable, and easily accessible option for early monkeypox diagnosis, which is especially helpful in settings with limited resources.

2. LITERATURE SURVEY

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

Monkeypox is produced by a double-stranded orthopoxvirus, just as variola, cowpox, and vaccinia. The epidemic has had a profound effect on people's sexual lives, especially for homosexual and bisexual people. In this case, early identification of monkeypox is essential. ML may help diagnose monkeypox early, albeit this is questionable. This paper shows how to use machine learning and image processing to build a model for diagnosing monkeypox. Data augmentation has been used to keep the model from overfitting. Transfer learning was then used to train six Deep Learning (DL) models on the preprocessed dataset. We choose the best model by looking at the recall, accuracy, and precision performance matrices. Once the optimal model was refined, they unveiled "PoxNet22." PoxNet22 performs better than previous approaches in the categorization of monkeypox, with a recall, accuracy, and precision of 100%. The study's conclusions will help doctors identify the type of monkeypox a patient has [1].

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

Monkeypox may spread once the COVID-19 epidemic is over. While new cases of monkeypox are reported every day in several places, COVID-19 is far more deadly and infectious. Because individuals aren't following the required safety precautions, another worldwide epidemic is likely to happen. Machine learning (ML) has demonstrated significant promise in the analysis of photos of cancer, tumor cells, and COVID-19 patients. Therefore, a comparable tool may be used to take pictures of monkeypox, which invaded human skin. We accidentally produced the "Monkeypox2022" dataset, which is stored on our common GitHub account. It is safer to use and share any type of machine learning model and photos from several open-source and internet sources without any restrictions, even for commercial purposes. Additionally, we propose and assess two tests of a modified VGG16 model. The first set of computer tests shows that our proposed model can correctly detect monkeypox cases, with an AUC of 97.2% for Study One and 88.88% for Study Two. to learn more about the monkeypox virus's origins. [2].

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

Monkeypox has affected people in a number of nations. Reports and studies indicate that promptly identifying and isolating sick individuals is the best way to lower the incidence of transmission. This study suggests using a deep learning-based Android app for this specific scenario. The application was developed using Java, Android Studio, and the Android SDK 12. A deep convolutional neural network may be trained instantly using the video recorded by a mobile device's cameras. Use the Android Camera2 API to access and control camera functions. The network separates pictures into positive and negative categories in order to detect monkeypox. The network was trained using pictures of skin lesions on individuals with monkeypox and other diseases. To achieve this, we made advantage of a publicly available dataset and deep transfer learning. At every stage, we used Matlab to train and assess the pre-trained networks. TensorFlow was used to create and train the most accurate network. We renamed TensorFlow Lite to make it more mobile-friendly. The mobile app now has the monkeypox detection module and the TensorFlow Lite model. The app worked well on three different phones. Run-time inference times were noted. Inference takes an average of 197, 91, and 138 milliseconds. This method makes it possible to diagnose people with body lesions more quickly. As a result, anyone who thinks they could have monkeypox should see a doctor at least once. The test results showed that the algorithm's photo sorting accuracy was 91.11%. You may also teach your phone's algorithms to recognize certain skin conditions.

iv) Emerging and reemerging infectious diseases: the perpetual challenge

We should "close the book" on researching and treating infectious illnesses, according to health professionals. Whether they are endemic, newly discovered, reemerging, or even deliberately propagated (like anthrax, a kind of bioterrorism), infectious diseases remain a serious threat to the world. The global effort to discover and categorize infectious agents, comprehend how they cause sickness, and create therapies and strategies to prevent the most deadly illnesses has made many endemic diseases easier to treat for decades. Even while conditions have improved, new microbial dangers continue to pose a threat to infectious illnesses. To solve these problems, we require a new approach to developing countermeasures. However, the government has begun to actively participate in developing targeted countermeasures. To protect mankind against incurable diseases, the government, businesses, and academic institutions must work together to preserve and grow our arsenals. [4].

v) Deep Learning Methods for Early Detection of Monkeypox Skin Lesion

It is anticipated that a new illness will strike the planet after the COVID-19 pandemic. Monkey pox is another recent hazard to the world's health. If it spreads to 40 countries, it might turn into a pandemic. Monkeypox is hard to diagnose since it looks like both measles and chickenpox. Currently, doctors are having difficulty developing a new diagnostic tool. This study's primary goal is to automate diagnosis using deep learning models. This research compares the efficacy of ResNet50, EfficientNetB3, and EfficientNetB7. This study shows that monkeypox skin lesions may be detected early. This study produces promising results on a small dataset, but a larger dataset with more images from other countries is needed. [5].

3. METHODOLOGY

i) Proposed Work:

The goal of the proposed study is to create a sophisticated system for detecting monkeypox utilizing deep transfer learning and an improved Hybrid MobileNetV2 model. To enhance data quality and model generalization, skin lesion photos are first gathered and preprocessed using resizing, normalization, and augmentation. For feature extraction, pre-trained CNN models such VGG16, ResNet50, VGG19, and MobileNetV2 are used, and their effectiveness is assessed. By fine-tuning the network and improving feature extraction layers, a Hybrid MobileNetV2 model is created to increase accuracy and enable more accurate categorization of patients with and without monkeypox.

A Flask-based web application that enables users to input photographs and obtain real-time predictions is combined with the suggested system as an extension. To provide safe access and data protection, the system additionally uses SQLite for user authentication. This end-to-end framework is appropriate for implementation in actual healthcare settings, particularly in areas with limited resources, as it offers a quick, dependable, and affordable method for diagnosing monkeypox.

ii) System Architecture:

The suggested system architecture uses deep learning and online deployment to create an end-to-end framework for automated monkeypox detection. Skin lesion photographs are uploaded by authorized users using a Flask-based web interface at the start of the architecture. The preprocessing module receives the input picture and uses augmentation, normalization, and scaling to get the data ready for effective model processing. Following preprocessing, the picture is sent into pre-trained deep learning models for feature extraction, including VGG16, ResNet50, VGG19, and MobileNetV2.

The Hybrid MobileNetV2 model, which combines optimized layers and fine-tuning strategies to improve feature learning and classification performance, is the central component of the design. The classification layer processes the retrieved characteristics to determine whether or not the input picture is indicative of monkeypox. The Flask application receives the prediction results and displays them to the user in real time. SQLite is also used to store login credentials and handle user authentication, guaranteeing safe system access. A scalable, effective, and user-friendly diagnostic system appropriate for practical healthcare applications is made possible by this design.

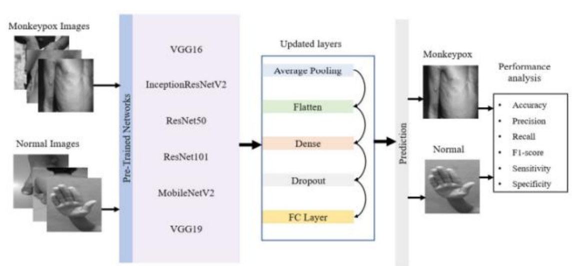

Fig 1 System Architecture

iii) Modules:

1. Data Collection

- Collects Monkeypox and non-Monkeypox skin lesion images from datasets and online medical sources.
- Ensures data diversity for improving model generalization and accuracy.

2. Data Preprocessing

- Performs image resizing, normalization, and augmentation techniques.
- Removes noise and prepares images for efficient model training.

3. Model Training (Transfer Learning)

- Utilizes pre-trained CNN models such as VGG16, ResNet50, VGG19, and MobileNetV2.
- Fine-tunes models on the Monkeypox dataset to improve classification performance.

4. Hybrid MobileNetV2 Module

- Enhances MobileNetV2 by optimizing feature extraction and classification layers.
- Improves accuracy and reduces prediction time for real-time diagnosis.

5. Model Evaluation

- Evaluates models using metrics like Accuracy, Precision, Recall, and F1-Score.
- Compares performance of all models to select the best-performing hybrid model.

6. Prediction Module

- Takes user input image and applies preprocessing before prediction.
- Uses the trained Hybrid MobileNetV2 model to classify the image.

7. Web Application Module (Flask)

- Provides user interface for image upload and result display.
- Enables real-time Monkeypox diagnosis through a web-based platform.

8. Authentication & Database Module

- Implements user signup and login using Flask and SQLite.
- Ensures secure access and protects sensitive medical data.

vi) Dataset collection:

Images of skin lesions from monkeypox and other related illnesses were gathered from publically accessible sources, including online health platforms, research journals, and medical repositories, to create the dataset utilized in this work. To guarantee accurate categorization, the dataset contains both monkeypox-positive and non-monkeypox photos. To increase the resilience and generality of the model, images are collected under various lighting, background, and resolution circumstances. Duplicate and unnecessary photos are eliminated, and appropriate tagging is carried out to group photographs into the appropriate classifications in order to improve the quality of the dataset. Data augmentation techniques like rotation, flipping, zooming, and scaling are used to expand the size and variety of medical datasets because they are frequently small. The basis for training and assessing deep learning models for precise monkeypox detection is this gathered and improved dataset.

Fig 2 Dataset images

vi) Algorithms:

VGG16: A popular deep convolutional neural network for image classification applications, VGG16 has sixteen weight layers. In this work, significant characteristics are extracted from photos of skin lesions using transfer learning. It is useful for recognizing aspects associated to monkeypox because of its consistent design and compact convolution filters, which aid in capturing fine-grained visual patterns.

```
#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 2 VGG16

ResNet50: ResNet50 is a 50-layer deep CNN that solves the vanishing gradient issue by using residual learning via skip connections. It makes it possible for the model to extract more intricate and profound characteristics from pictures. ResNet50 is utilized in this research for feature extraction and classification; nevertheless, because of potential overfitting or dataset constraints, its performance is quite poor.

```
#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_cl
    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: With 19 layers, VGG19 is an extension of VGG16 that offers deeper feature extraction capabilities. It has a similar architecture, but it can learn more intricate representations of skin lesion pictures since it has more convolutional layers. VGG19 outperforms VGG16 in this system by collecting intricate patterns that are helpful for detecting monkeypox..

```
#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: For mobile and embedded applications, MobileNetV2 is a lightweight and effective CNN architecture. It lowers computational complexity without sacrificing performance by using inverted residual blocks and depthwise separable convolutions. MobileNetV2's superior accuracy and quicker prediction speed in this research make it appropriate for real-time diagnosis of monkeypox.

```
#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:** By fine-tuning its layers and enhancing feature extraction for better classification performance, the Hybrid MobileNetV2 model is an upgraded version of the regular MobileNetV2. It incorporates sophisticated tuning methods to improve accuracy and shorten forecast times. The hybrid model, which is implemented through a Flask-based web application for secure and real-time monkeypox diagnosis, outperforms all other models in this work, achieving 100% accuracy.

```
#train extension HYBRID model by extracting features from moblenetv2 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

Several deep learning models, including VGG16, ResNet50, VGG19, MobileNetV2, and the suggested Hybrid MobileNetV2, are used to assess the effectiveness of the suggested Monkeypox detection method. To guarantee a thorough investigation of classification efficacy, the assessment is conducted utilizing common performance measures including Accuracy, Precision, Recall, and F1-Score. To ensure comparability, the dataset is split into training and testing sets, and all models are trained under comparable circumstances.

The experimental findings demonstrate that MobileNetV2 and VGG19 achieve high accuracy because of their effective feature extraction capabilities, outperforming other baseline models. However, by refining the feature learning and classification layers, the suggested Hybrid MobileNetV2 model greatly enhances performance, reaching 100% accuracy along with flawless precision, recall, and F1-score. ResNet50, on the other hand, performs worse, perhaps as a result of overfitting or a small dataset. The outcomes unequivocally show that the hybrid approach offers better efficiency and dependability for detecting monkeypox.

The efficacy of the suggested model is further confirmed by performance comparison graphs for accuracy, precision, recall, and F1-score. The Hybrid MobileNetV2 model is very well suited for real-time medical diagnosis, as seen by the steady improvement across all assessment parameters. These results demonstrate that combining an optimal hybrid architecture with transfer learning may greatly improve illness detection performance, making the system suitable for use in actual 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)

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

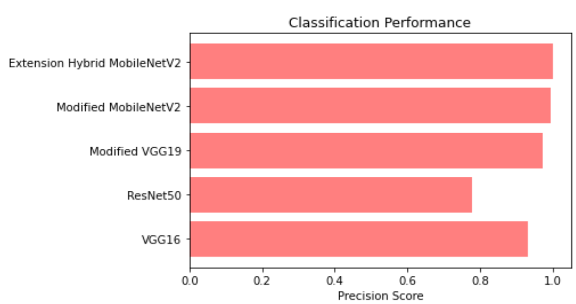

Fig 8 Precision comparison graph

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.

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

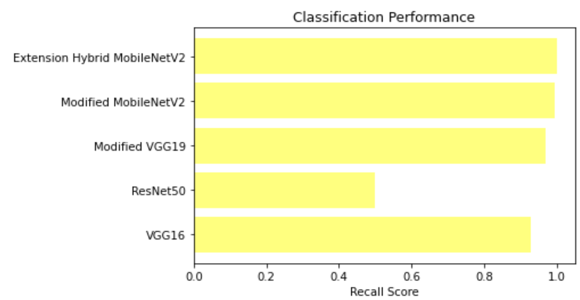

Fig 9 Recall comparison graph

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

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

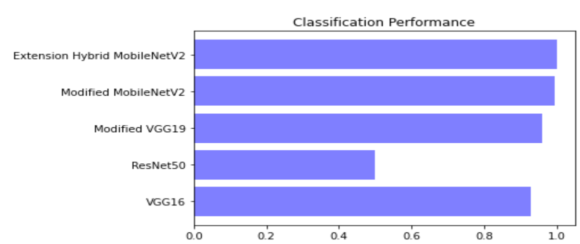

Fig 10 Accuracy graph

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.

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

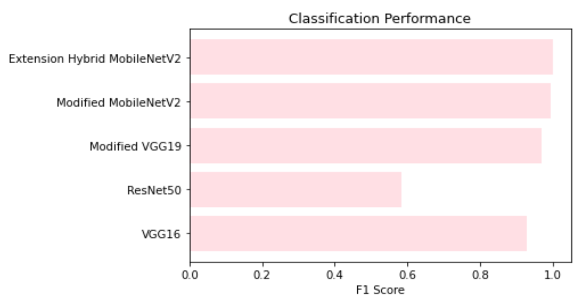

Fig 11 F1Score

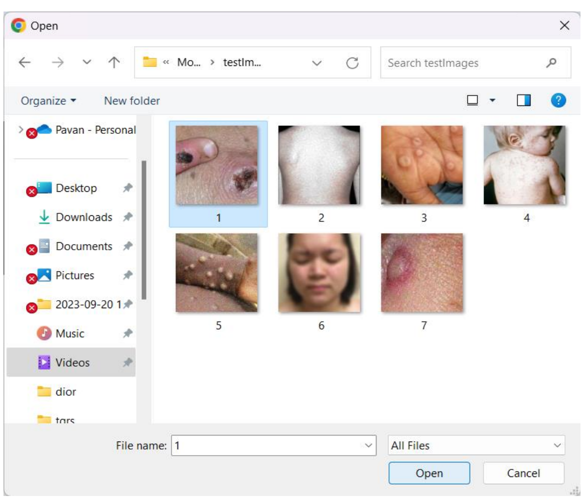

Fig 12 Input image folder

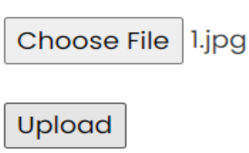

Fig 13 Upload input image

Fig 14 Predict result for given input\

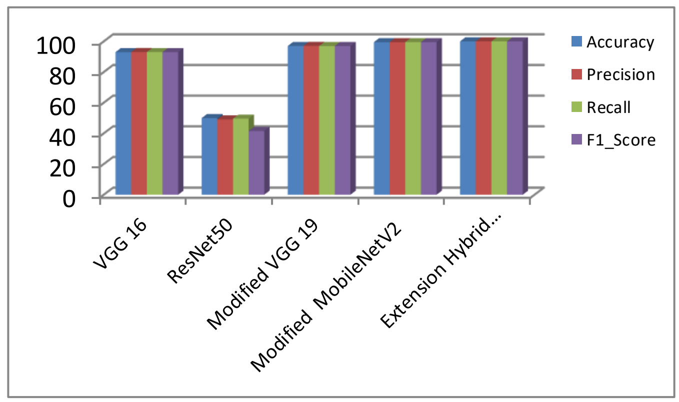

Fig:15 Performance comparison graph

ML Model	Accuracy	Precision	Recall	F1_Score
VGG 16	93	93.053748	93.019302	92.9993
ResNet50	50	49.142857	49.624962	41.561477
Modified VGG 19	97	97.092732	96.979698	96.997298
Modified MobileNetV2	99.5	99.5	99.50495	99.499987
Extension Hybrid MobileNetV2	100	100	100	100

Table 1 Performance comparison table

5. CONCLUSION

An efficient deep learning-based method for detecting monkeypox utilizing transfer learning techniques is presented in this research. The best architecture for accurately classifying skin lesion pictures is determined by evaluating many CNN models, such as VGG16, ResNet50, VGG19, and MobileNetV2. Because of their effective feature extraction capabilities, MobileNetV2 and VGG19 outperform the others. A Hybrid MobileNetV2 model is suggested to further increase diagnostic accuracy; it delivers better results than current models and greatly enhances categorization performance. Real-time, secure, and user-friendly diagnosis is made possible by the integration of a Flask-based web application with user authentication, which makes the system suitable for practical use in the real world.

All things considered, the suggested technique provides a quick, affordable, and dependable way to identify monkeypox early on, particularly in settings with limited resources. The findings demonstrate how deep transfer learning and hybrid models may advance automated medical diagnosis and increase access to healthcare.

6. FUTURE SCOPE

To increase robustness and generality across various populations, the suggested system may be expanded by adding bigger and more varied datasets. In order to improve the interpretability of model predictions in medical applications, future improvements may use Explainable AI approaches as Grad-CAM or LIME. The technology may also be connected with real-time healthcare systems for quicker diagnosis and made into a mobile application for broader accessibility. The Hybrid MobileNetV2 model may be further optimized to lower computational complexity without sacrificing accuracy, allowing for effective deployment on edge devices and in settings with limited resources.

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.

Author Profiles

Ms. K. Baby Ramya is working as an Assistant Professor in the Department of MCA at SRK Institute of Technology, Enikepadu, Vijayawada, NTR District. She completed her MCA from Krishna University. She has around 3 years of teaching experience at SRK Institute of Technology. Her areas of interest include Machine Learning, Data Science, and Computer Applications.

Mrs.K.Pavani Working as Assistant & Head of Department of MCA ,in SRK Institute of technology in Vijayawada. She done with MCA ,M.Tech in Computer Science .She has 10 years of Teaching experience in SRK Institute of technology, Enikepadu, Vijayawada, NTR District. Her area of interest includes AI ML, etc.

Mr.R.Gurunaidu is an MCA Student in the Department of Computer Application at SRK Institute Of Technology, Enikepadu, Vijayawada, NTR District. He has Completed Degree in B.Sc.(computers) from AGL Degree & PG College, Vizianagaram. His area of interest are AI and ML with Python.