

Feature Fusion of Ensemble Deep Learning Models for Classification of Monkeypox Disease

Nidhi Chauhan¹, Dr. Sunita Yadav², Dr. Alok Singh Chauhan³

1. Department of Computer Science and Engineering, Galgotias University, Gautam Buddha Nagar, Uttar Pradesh

2. Department of Computer Science and Engineering, Galgotias University, Gautam Buddha Nagar, Uttar Pradesh

3. Department of Computer Science and Engineering, Galgotias University, Gautam Buddha Nagar, Uttar Pradesh

Article History:

Received: 19-11-2024

Revised: 07-12-2024

Accepted: 14-01-2025

Abstract:

Outbreaks of human monkeypox have been reported in many nations recently, including India. Studies and reports indicate that to reduce the spread of the infection, it is crucial to identify and isolate affected individuals as soon as possible. The purpose of this work is to use integration of pre-trained deep learning models for extracting the features from monkeypox image dataset and then combine them for detecting and diagnosing Monkeypox disease. Combining features provides a rich and comprehensive representation of the input data, enabling the classifier to make more reliable predictions. In our study we used 5 feature fusion techniques to obtain and select best feature representations. Each method offers different ways of fusing information. Two publicly available datasets are utilised: Monkeypox Skin Image Dataset (MSID), Monkeypox Skin Lesion Detection (MSLD). Fusing features from models like pre-trained models effectively combines their strengths, improving generalization, accuracy, and robustness in Monkeypox detection. The model's performance was evaluated using six metrics: accuracy, recall, precision, F1-score, ROC, and AUC. Our experimental results demonstrate that the combined features from pre-trained DenseNet121 and MobileNetV2 models using multiplicative fusion method have the best classification accuracy of 95.56% on MSLD and 91.67% on MSID.

Keywords: Monkeypox disease, Deep Learning Methods, Integrated model, MobileNetV2, DenseNet121, Fused features.

1. Introduction

Monkeypox has recently affected over 16,000 individuals across 75 countries, raising significant global health concerns. Similar to several disease, like COVID-19, which infected 23 distinct animal species, monkeypox has the potential to spread from domestic to wild animals, endangering the health of all. Although human-to-human contact is the primary cause of the current monkeypox outbreak, the disease is known to have an animal origin and can thus be transmitted to specific animals. Various wild mammals, including dormice, rope squirrels, tree squirrels, Gambian pouched rats, and non-human primates, have been found to be vulnerable to the monkeypox virus [[23]]. Identifying spillover events is made more difficult by the fact that while some of these species display disease signs and symptoms, others may not display any external or apparent indications. Also, the skin lesions of the disease are very similar to rashes of smallpox, measles and chickenpox. As the number of cases of monkeypox increased, numerous initiatives, plans, and countermeasures were put in place by nations worldwide to slow the virus's spread. The United States procured 500,000 doses of the smallpox vaccine, Canada offered vaccinations to high-risk groups, French and Danish health authorities

advocated for the distribution of vaccines to adults affected by the virus, Germany recommended vaccinations for high-risk populations, and the United Kingdom advised self-isolation for all virus-infected individuals. These efforts included the implementation of lockdown measures in Belgium [[11]][[9]]. While writing this research paper, 2 cases in Kerala, India, were reported. As its spread is nationwide, it is necessary to detect this disease on early stages. Deep learning (DL) plays a pivotal role in health informatics, excelling in feature extraction and data classification [2]. Representation learning, the foundation of deep learning techniques, entails building nonlinear modules layer by layer to reach higher representation levels. The representation is changed by each layer from one form to another, eventually producing a more abstract representation that makes it easier for a feature set to be generated automatically [[3]][[4]]. Automated feature generation without human interaction offers substantial benefits in health informatics. Despite their encouraging results, a crucial issue is the lack of comprehension and interpretability of these models' decision-making procedures. The issue of establishing trust and acceptance among professionals is posed by this lack of transparency. Addressing these issues, the main objectives of this work are described as:

- To develop an efficient and accurate diagnostic model for monkeypox disease by integrating pre-trained deep learning models and leveraging advanced feature fusion techniques.
- Explore various feature fusion techniques (e.g., concatenation, additive fusion, multiplicative fusion) to determine the best approach for combining features extracted from the models.
- Analyse the advantages and limitations of each fusion method to identify the most effective technique for improving classification accuracy.
- Evaluate the model using six performance metrics (accuracy, precision, recall, F1-score, ROC, and AUC) to ensure reliability and generalizability.
- Conduct comparative analysis on two publicly available datasets (MSID and MSLD) to validate the model's effectiveness.

This research aims to address the challenges of early and accurate detection of monkeypox disease, contributing to better disease management and containment strategies.

2. Related work

Kinjal A. Patel et al., compares several deep learning (DL) and machine learning (ML) models for monkeypox detection. They highlight the ways in which contemporary data analytics may strengthen surveillance efforts, expedite intervention, and increase early detection in the fight against monkeypox epidemics [[5]]. The study contrasts sophisticated deep learning models like Convolutional Neural Networks (CNN) and Extended Convolutional Neural Networks (ECNN) with more conventional machine learning techniques like Decision Trees, Support Vector Machines (SVM), and Random Forests[20]. The study's ECNN model identified monkeypox with an accuracy of 88.10%. Compared to machine learning methods, it performed better in the classification of infected individuals. The study concludes that deep learning methods, in particular CNN and ECNN, significantly enhance the ability to identify monkeypox. Krishna Raj Chadaga et al. conduct a thorough evaluation of the application of Artificial Intelligence (AI) techniques in the field of Monkeypox (Mpox) epidemiological modelling, therapy, and diagnosis [[6]]. The comprehensive study found 34 studies that used AI methods to fight malaria. MobileNet, VGG-16, ResNet, and InceptionNet are the AI models that are

most frequently employed in detection. This paper offers a thorough analysis of how artificial intelligence (AI) is furthering the study of monkeypox and developing medical treatments, highlighting its use in clinical settings as well as in understanding the transmission of the virus and public opinion. Possible uses of AI in telemedicine and cloud-based Mpox management systems are explored as potential future directions. Katy Brown et al. emphasizes the need for improved diagnostic tools, vaccines, and antiviral treatments to prevent future monkeypox outbreaks, especially considering environmental changes that increase contact with reservoir hosts. Strengthening surveillance systems, improving diagnostic capacity, and educating populations in endemic areas about zoonotic risks are crucial. Pre-event vaccination of healthcare workers and high-risk individuals is recommended by health agencies like the CDC [[7]]. Sahin, V.H. et al. proposes an Android mobile application that classifies skin lesions and determines if they are monkeypox-related using deep learning. By offering users a prompt, non-invasive preliminary diagnosis, the app hopes to encourage those who are infected to seek expert medical assistance as soon as possible. The system's fundamental component is a deep learning model that classifies images using pre-trained networks (such MobileNetV2 and EfficientNetB0) on MSLD dataset. F1 score, accuracy, and precision are among the common measures used to assess the performance of the system. With an accuracy rate of 91.11%, the MobileNetV2 model produced the best results and is a par with other cutting-edge techniques in this field [[8]]. Irmak, M.C. et al. investigated the accuracy metrics of three approaches for classifying images from the Monkeypox Skin Image Dataset, which was made publicly available in 2022[[21]]. The previously trained CNN networks MobileNetV2, VGG16, and VGG19 were used for this task. MobileNetV2 achieved the best performance results, with 91.38% accuracy, 90.5% precision, 86.75% recall, and 88.25% f1 score. The accuracy of the VGG16 and VGG19 methods, respectively, was 83.62% and 78.45% [[9]]. Shivangi Surati, et al. suggests a deep learning-based method for categorising measles, chickenpox, and monkeypox to reduce the difficulty of identifying the disease. The reason for this is the overlapping visual symptoms of monkeypox making it difficult to differentiate it from other diseases. An attention model merged with well-known convolutional neural networks (CNNs) like InceptionV3, EfficientNet, and VGG16 is introduced in the paper as the Squeeze and Excitation Network (SENet) [[10]]. By strengthening the feature extraction capabilities and concentrating on pertinent areas of the image, the model's performance is intended to be improved. This research is the first to use SENet for the classification of monkeypox diseases. MonDiaL-CAD, a computer-aided diagnostic (CAD) tool intended to improve the speed and accuracy of monkeypox detection, is introduced in this study [[11]]. The technique leverages several pre-trained CNNs (Convolutional Neural Networks) to enhance classification outcomes by amalgamating the advantages of several CNN designs for feature extraction. To find the optimal mix of these deep feature sets, MonDiaL-CAD pulls features from eight distinct CNN models (such as ResNet-18, MobileNet, and Xception) and applies a forward search technique. This technique enhances diagnosis accuracy, minimises feature space, and maximises model performance. Deep features are fused using a DWT approach, which reduces their size and yields a spatial-time-frequency representation. The approach reduces the dimensions of the fused feature set by applying an entropy-based feature selection mechanism after the features have been fused. Using fusion classifiers like Bagging, Rotation Forest, and Random Subspace, the actual classification of monkeypox against other diseases is done in the last stage. Using the MSID dataset, the MonDiaL-CAD achieves 97.1% accuracy, while using the MSLD dataset, it achieves 98.7%

accuracy. Uysal F. developed a hybrid deep learning model that combines long short-term memory (LSTM) networks and convolutional neural networks (CNNs) for the purpose of classifying skin disease images, with a particular emphasis on monkeypox detection. Initially, seven models were trained independently: CSPdarkNet, InceptionV4, MnasNet, MobileNetV3, RepVGG, SE-ResNet, and Xception. With respect to performance, RepVGG and MnasNet performed best. A test accuracy of 87% and a Cohen's kappa score of 0.8222 were obtained by combining the two top CNN models (RepVGG, MnasNet) with an LSTM model, indicating increased performance. In terms of accuracy, precision, and recall, the hybrid deep learning model performed better than the individual models, indicating the importance of integrating CNN and LSTM for image classification [[12]]. By combining CNN (Convolutional Neural Network) and Vision Transformer models to create an automated classification system, the study tackles the rising problem of diagnosing skin diseases, especially monkeypox and other lesions [[13]]. It aims to offer dermatologists rapid, accurate, and non-invasive methods of diagnosis to help them classify different types of skin lesions. Additionally, the study used fusion learning and CNN models (such as DenseNet201) to improve performance by combining the advantages of various models (CNN and Vision Transformer). The final fusion model performed well in terms of precision, recall, F1-score, and Jaccard metrics, with an accuracy of 81.91%. Merely applying the Vision Transformer, 80.66% accuracy was attained. The system's performance was further enhanced by fusion learning with the Bagging technique, demonstrating the combined strengths of the CNN and Vision Transformer models.

3. Methods

This section is divided into 5 parts: data collection, deep learning models used and selection of best model, feature fusion techniques with advantages and disadvantages of each technique, the proposed method and the evaluation parameters used to predict and classify Monkeypox disease.

3.1 Dataset

Two publicly available datasets namely, MSID and MSLD have been used in this study [[22]]. The dataset summary is shown in Table I. In MSID dataset two categories are utilized i.e. “Monkeypox” and “Chickenpox + Measles” combined which act as normal class. In MSLD two categories are “Monkeypox” and “Normal”. The images in the datasets are of 224x224x3 size. Some sample images are shown in Figure 1. The dataset size was increased using data augmentation techniques such as shear, rotation, and scaling.



Figure 1. Sample images of monkeypox from datasets used in this study.

Table I. Dataset distribution

<i>Dataset</i>	<i>Label</i>	<i>Train</i>	<i>Test</i>	<i>Validate</i>	<i>Total</i>
MSID[[15]]	Monkeypox	223	56	0	279
	Chickenpox + Measles	158	40	0	198
	Total	381	96	0	469
MSLD[[16]]	Monkeypox	980	20	168	1168
	Normal	1162	25	252	1439
	Total	2142	45	420	2607

3.2 Deep Learning models for Feature extraction

Two best-performing models among five deep learning models have been chosen for feature extraction. The models applied to each dataset separately are ResNet50, VGG16, DenseNet121, MobileNetV2, and InceptionV3. Out of these models, DenseNet121 and MobileNetV2 performed well on MSLD and MSID, respectively, in terms of accuracy. Hence, we utilized the advantages of these two commonly known transfer deep learning models for performing feature extraction. The feature extraction task is performed individually by pre-trained models. The extracted features from both models are then combined using several fusion techniques for the further classification process.

DenseNet121 architecture: DenseNet121 is a deep learning model designed to enhance feature propagation and reuse by connecting each layer to every other layer within a dense block [[24]]. DenseNet121 is composed of four dense blocks with transition layers in between as shown in Table II. Below is an overview of its architecture:

Input Layer:

- Input images of size $224 \times 224 \times 3$.

Convolutional Layer:

- 7×7 convolution with stride 2 and 64 feature maps.
- Followed by batch normalization, ReLU, and 3×3 max pooling.

Four Dense Blocks:

- Dense Block 1: Contains multiple layers, each receiving inputs from all previous layers.
- Transition Layer 1: Compresses features and passes them to the next dense block.
- This structure is repeated for Dense Blocks 2, 3, and 4.

Global Average Pooling and Classification:

- A 7×7 global average pooling layer.
- The final layer is a fully connected layer with softmax activation function.

Table II. Dense block and transition layer of DenseNet121 architecture.

Layer Type	Output Size	Details
Convolution	112 × 112	7 × 7, stride 2, 64 filters
Max Pooling	56 × 56	3 × 3, stride 2
Dense Block 1	56 × 56	6 layers, growth rate 32
Transition Layer 1	28 × 28	1 × 1 conv, 2 × 2 avg pooling
Dense Block 2	28 × 28	12 layers, growth rate 32
Transition Layer 2	14 × 14	1 × 1 conv, 2 × 2 avg pooling
Dense Block 3	14 × 14	24 layers, growth rate 32
Transition Layer 3	7 × 7	1 × 1 conv, 2 × 2 avg pooling
Dense Block 4	7 × 7	16 layers, growth rate 32
Global Avg Pooling	1 × 1	-
Fully Connected	1 × 1	Softmax, 1000 classes (ImageNet)

MobileNetV2 architecture: MobileNetV2 is a lightweight convolutional neural network architecture designed for resource-constrained environments such as mobile and embedded devices. It builds on the success of MobileNetV1 by introducing inverted residual blocks and linear bottlenecks for efficient computation and higher accuracy. [[25]]. The backbone of MobileNetV2 consists of inverted residual blocks shown in Table III.

Architecture overview is shown below:

Input Layer:

- Accepts an image of size 224×224×3.

Initial Convolution:

- A standard 3×3convolution with stride 2, outputting 32 filters.

Bottleneck Blocks:

- The core of MobileNetV2 consists of a series of bottleneck residual blocks. Each block is defined by:

- Expansion: Expands the input using 1×1 convolution.
- Depthwise Convolution: Applies 3×3 convolution with stride 1 or 2.
- Linear Projection: Reduces the expanded dimensions back to a smaller bottleneck size.

- Blocks are parameterized by:

- Input and output channels.
- Expansion factor (commonly 6).
- Stride.

Final Convolution and Classifier:

- After the bottleneck blocks, a 1×1 convolution with 1280 filters is applied.
- Followed by global average pooling and a fully connected softmax layer.

Table III. Inverted residual blocks of MobileNetV2 architecture.

Stage	Input Size	Operation	Output Channels	Stride
0	224 × 224	3 × 3, Conv2D, 32 filters	32	2
1	112 × 112	Bottleneck, expansion=1	16	1
2	112 × 112	Bottleneck, expansion=6	24	2
3	56 × 56	Bottleneck, expansion=6	24	1
4	56 × 56	Bottleneck, expansion=6	32	2
5	28 × 28	Bottleneck, expansion=6	32	1
6	28 × 28	Bottleneck, expansion=6	32	1
7	28 × 28	Bottleneck, expansion=6	64	2
8	14 × 14	Bottleneck, expansion=6	64	1
9	14 × 14	Bottleneck, expansion=6	96	1
10	14 × 14	Bottleneck, expansion=6	96	1
11	14 × 14	Bottleneck, expansion=6	160	2
12	7 × 7	Bottleneck, expansion=6	160	1
13	7 × 7	Bottleneck, expansion=6	320	1
14	7 × 7	1 × 1, Conv2D, 1280 filters	1280	1
15	1 × 1	Global Average Pooling	1280	-
16	1 × 1	Fully Connected, Softmax	Number of Classes	-

3.3 Feature fusion techniques

The extracted features are fusion together using different fusion approaches. After the combined features, classification is done by fully connected layers of CNN. We can combine features in various ways like concatenating the features, adding features, multiplying features etc [[14]]. The best option generally relies on the task and dataset. Each method offers a different way to fuse information. The following are the primary techniques used in this research for joining extracted features:

Simple Concatenation: Directly concatenating the feature vectors from two or more models. It is the most widely used technique. The entire feature representation from both models is maintained using this way. The function for the concatenation of two features vectors is given by equation $\text{Concatenate}(\text{model}_1, \text{model}_2)$ (1).

$$\text{Concatenate}(\text{model}_1, \text{model}_2) \tag{1}$$

Weighted Concatenation: To regulate each model's contribution, you can provide the features that were taken from pre-trained models distinct weights. When certain properties of one model are more significant than those of the other, this can be useful. Function for this concatenation is given in equation $\text{Concatenate}([0.7*\text{model}_1, 0.3*\text{model}_2])$ (2).

$$\text{Concatenate}([0.7*\text{model}_1, 0.3*\text{model}_2]) \tag{2}$$

Additive Fusion (Element-wise Addition): It can add the feature vectors elementwise rather than concatenating. For this to work, the dimensions of both feature vectors must match. You must apply a transformation (such as a Dense layer) if the feature vector sizes differ to guarantee that they are of the same size. Function for element wise addition is given in equation $\text{Add}(\text{model}_1, \text{model}_2)$ (3).

$$\text{Add}(\text{model}_1, \text{model}_2) \tag{3}$$

Element-wise Multiplication (Hadamard Product): Multiply the feature vectors elementwise. Interactions between the features are captured using this technique. Element-wise multiplication of features is given by equation $\text{Multiply}(\text{model}_1, \text{model}_2)$

$$\tag{4}.$$

$$\text{Multiply}(\text{model}_1, \text{model}_2) \tag{4}$$

Gated Fusion (Attention Mechanism): To choose the amount of information to extract from each model, you can apply either an attention mechanism or a gating function. The merging of pre-trained models' features is managed by a learned gating layer. The function for this type of concatenation is given by equation $\text{Multiply}([\text{gate}, \text{model}_1]) + \text{Multiply}([(1 - \text{gate}), \text{model}_2])$

$$\tag{5}.$$

$$\text{Multiply}([\text{gate}, \text{model}_1]) + \text{Multiply}([(1 - \text{gate}), \text{model}_2]) \tag{5}$$

Feature Averaging: The feature vectors are averaged rather than concatenated. This is a straightforward method of merging two models so that each makes an equal contribution. Average function of this concatenation is given by equation $\text{Average}(\text{model}_1, \text{model}_2)$

$$\tag{6}.$$

$$\text{Average}(\text{model}_1, \text{model}_2) \tag{6}$$

The Pros and Cons of each method is shown in **Error! Reference source not found.** with the functions available in Keras library.

Table IV. Feature fusion techniques with advantages and disadvantages.

<i>Technique</i>	<i>Pros</i>	<i>Cons</i>
<i>Simple Concatenation</i>	Retains the maximum information possible from both networks.	It can cause feature vectors to become highly dimensional, which could increase training time and need more memory.
<i>Weighted Concatenation</i>	Permits giving priority to more relevant features.	Requires tuning of the weights, which might be tricky.
<i>Additive Fusion (Element-wise Addition)</i>	Easier to control in terms of memory and processing power. Decreases the dimensionality of the combined feature space.	By direct addition, some detailed feature representations might be lost.
<i>Multiplication</i>	Results in smaller feature dimensions compared to concatenation.	Might lose the significance of each attribute when multiplied directly.
<i>Gated Fusion (Attention Mechanism)</i>	Determines the attributes that are most important dynamically.	More sophisticated in terms of calculation.
<i>Feature Averaging</i>	Easy to understand and computationally effective.	The process of averaging may lose some crucial details. Less flexible

		than concatenation and other similar techniques.
--	--	--

3.4 Proposed Methodology

After choosing high accuracy performance deep learning models i.e. pre-trained models for feature extraction and analysing all feature fusion techniques, we have followed the methodology shown in Figure 2 for classifying image datasets into ‘Monkeypox’ or ‘Normal’ category. The training dataset have been passed on pre-trained models separately to extract feature maps. The obtained feature maps are fused using the specified fusion methods. The best method to fuse features is being chosen by analysing the accuracy as well as the ease of representation learning. The fused feature maps are then used to classify the Monkeypox images and normal images.

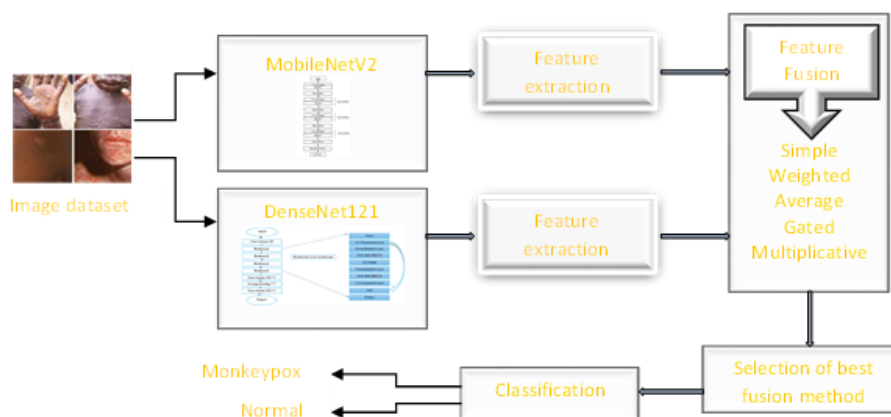


Figure 2. A visual flowchart of proposed model.

The algorithm for the process followed for our work is shown next in Table V. Image width and height are set to be 224. Pre-trained deep learning models are loaded. Feature extraction is performed by these models individually and then concatenation of the features is performed using various concatenation functions.

Table V. Algorithm of methodology adopted.

<p>Algorithm</p> <ol style="list-style-type: none"> 1. Load Dataset img_width, img_height = 224, 224 2. Import Libraries TensorFlow, Keras, or PyTorch. 3. Load Pre-trained MobileNetV2 and DenseNet121 models mobile_net = MobileNetV2() dense_net = DenseNet121() 4. Concatenate the Features Concatenate()([mobile_net_output, dense_net_output]) 5. Add Dense Layers for Classification FC+ softmax layer 6. Build final model Model(inputs=[mobile_net.input, dense_net.input], outputs=output) 7. Model compilation model.compile() 8. Model training model.fit()
--

3.5 Evaluation parameters

For evaluation of the proposed model, 8 well-known metrics are used in our study:

- *Accuracy*: the overall number of correctly identified instances across all instances.
- *Sensitivity*: (also known as recall): ratio of true positive instances over the actual number of positive instances.
- *Specificity*: ratio of “true negative” values in all cases that do not belong to this class.
- *Precision*: ratio of true positive instances over the total number of positive instances.
- *F1score*: harmonic mean of precision and recall [[17]].
- *ROC*: Receiver Operating Characteristic (ROC) curve illustrates how effectively a binary classifier model works at various threshold settings. For every threshold setting, it plots the true positive rate (TPR) against the false positive rate (FPR). Area Under the Curve (AUC) score help to evaluate the model performance ranging from 0 to 1 [[18]].

4. Results

We have compared various methods of combining features extracted from two CNN models i.e. DenseNet121 and MobileNetV2. We have chosen these two models among 5 models (ResNet50, VGG16, DenseNet121, MobileNetV2, InceptionV3) by comparing the accuracy performance and considerations like computational cost, memory usage etc. MobileNetV2 is ideal for lowering resource usage because it is very memory-efficient and computationally light. Because of its depth wise separable convolutions, MobileNetV2 is efficient at extracting low-level and mid-level features. DenseNet121 minimises the number of parameters and avoids the vanishing gradient issue by promoting feature reuse through dense connections. Because of its densely connected layers, DenseNet121 is excellent at extracting deeper and higher-level features. By combining them, a rich and varied feature representation is produced, enhancing model performance without putting excessive computational burden. The combined features extracted from both the models are then used for classification of monkeypox on two datasets used. For combining features, 5 feature fusion techniques (simple, weighted, average, multiplicative, additive and gated) were utilised. Out of these, multiplicative feature fusion performs well in terms of accuracy, recall and F1-score. Although 6 evaluation metrics have been used in our study: test accuracy, F1-Score, precision, recall, ROC, AUC score. Multiplicative feature fusion often yields higher accuracy in combining features from two deep learning models because it focusses feature interactions on a way that shows mutual relevance and interdependencies. Noise is reduced when weak or irrelevant features from one model are multiplied by strong features from the other model. This method creates a richer feature space compared to simple concatenation, which merely aggregates features. It maintains the dimensionality of the feature space consistent with the original features, in contrast to concatenation, which increases the dimensionality of the feature space (perhaps resulting in overfitting or increased computing costs). As a result, the model maintains excellent performance while being computationally efficient. Even while multiplicative fusion provides benefits, it might not be the ideal option all the time. It necessitates

Careful feature dimension matching and may suffer from poorly scaled or biased features in one model. To ensure successful fusion, normalisation and scaling algorithms are frequently required.

Result of all feature fusion methods on MSLD dataset is shown in Table VI. By the experiments it has been observed that multiplicative fusion method performs better than other state-of-art methods on both datasets with test accuracy of 95.56% on MSLD and 91.67% on MSID.

Table VI. A comparison of feature fusion methods on MSLD dataset.

Metrics	Simple Concatenation	Weighted Fusion	Average Fusion	Additive Fusion	Gated Fusion	Multiplication Fusion
Test Accuracy (%)	88.89	86.67	93.33	91.11	77.78	95.56
Precision	1.00	1.00	0.958	0.862	0.714	0.926
Recall	0.80	0.76	0.92	1.00	1.00	1.00
F1-Score	0.889	0.864	0.939	0.926	0.833	0.962
AUC Score (%)	93.00	88.00	93.00	90.00	75.00	95.00

A graphical comparison of multiplicative fusion method with other methods on MSLD datasets is shown in Figure. .

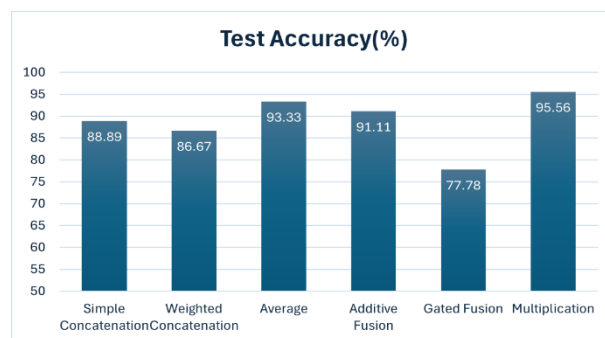


Figure. 3. Results of best performed fusion method on MSLD dataset.

Confusion matrices of fusion methods on MSLD dataset is shown in Figure. **Figure. .** Two categories classification is shown. Multiplicative method has 18 true positive (TP) values and 25 true negatives (TN) in MSLD dataset. True positives indicate correctly predicted Monkeypox cases and true negatives indicates correctly predicted non-Monkeypox (Chickenpox+Measles) or normal cases. High TP and TN values indicate that the model is correctly classifying most cases.

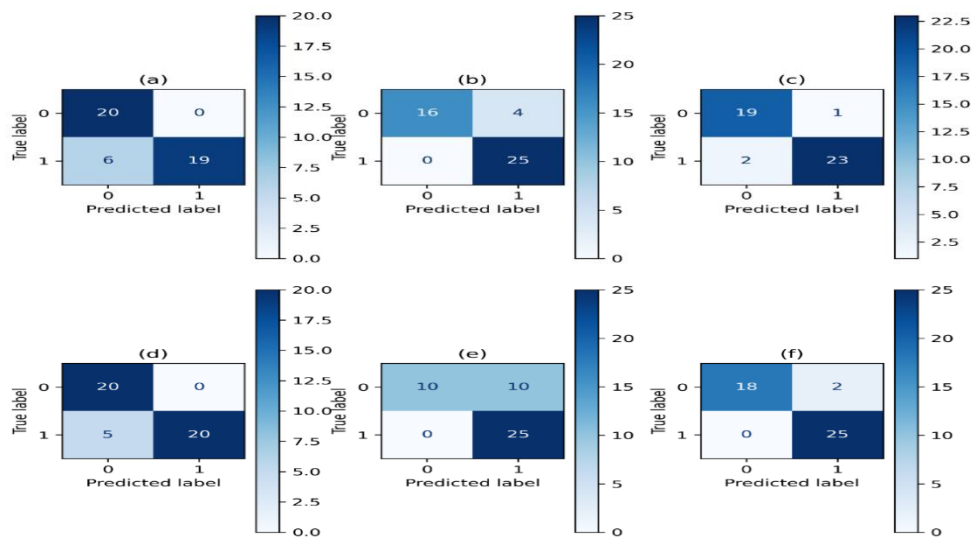


Figure 4. Confusion matrices of fusion methods on MSLD (a) weighted (b) additive (c) average (d) simple (e) gated (f) multiplicative

Next shown in **Figure** are the ROC curves of all fusion methods on MSLD dataset. Multiplicative fusion method has the highest AUC score of 0.95. An AUC closer to 1 indicates a model with excellent performance.

Figure

Figure 5. ROC graphs of all fusion methods on MSLD (a) weighted (b) additive (c) average (d) simple (e) gated (f) multiplicative

Next shown are the results obtained on MSID dataset. A tabular representation of all feature fusion methods on MSLD dataset is shown in Table VII.

Table VII. A comparison of feature fusion methods on MSID dataset.

Metrics	Simple Concatenation	Weighted Fusion	Average Fusion	Additive Fusion	Gated Fusion	Multiplicative Fusion
Test Accuracy (%)	83.33	91.65	89.58	85.42	86.46	91.67
Precision	0.833	0.881	0.857	0.795	0.829	0.90
Recall	0.75	0.925	0.90	0.875	0.85	0.90
F1-Score	0.789	0.902	0.878	0.833	0.84	0.90
AUC Score (%)	82.00	92.00	90.00	86.00	86.00	91.00

A graphical comparison of multiplicative fusion method with other methods on MSID datasets is shown in Figure .

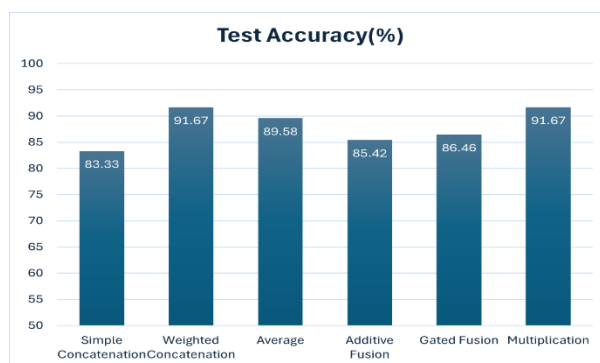


Figure 6. Results of best performed fusion method on MSID dataset.

Confusion matrices of fusion methods on MSID dataset is shown in Figure . The confusion matrix of multiplicative method has 52 true positives and 36 true negatives. It shows that it correctly identifies the Monkeypox cases and Normal cases when compared to other state-of-art fusion methods.

Figure

Figure 7. Confusion matrices of fusion methods on MSID (a) weighted (b) additive (c) average (d) simple (e) gated (f) multiplicative

Further below shown in **Error! Reference source not found.** are the ROC curves of all fusion methods on MSID dataset. AUC is a single scalar value that summarizes the overall performance of a binary classification model. It measures the ability of the model to distinguish between the positive and negative classes. Multiplicative fusion method has the AUC score of 0.91.

Error! Reference source not found.

Figure 8. ROC graphs of all fusion methods on MSID (a) weighted (b) additive (c) average (d) simple (e) gated (f) multiplicative

5. Conclusion

A rich and thorough representation of the input data is provided by feature fusion, which helps the classifier provide more accurate predictions. Each fusion method has its advantages and disadvantages depending on the task and dataset used. Feature fusion leverages individual models to extract diverse and complementary features from input images. When dealing with complicated or noisy data, the model's overall performance may be enhanced by this diversity of feature sources. Through the integration of characteristics from several architectures, the fusion model gains an enhanced ability to generalize across unknown data. Combining several feature sets gives the classifier access to richer, more insightful data, which improves prediction accuracy. This is especially crucial for tasks involving the diagnosis of diseases because even tiny increases in accuracy can have a significant therapeutic impact. The combination of features retrieved by pre-trained models enhances the model's capacity to distinguish across related classes. The fusion of features captures distinct image characteristics, including texture, edges, and fine details. Element-wise multiplication fusion captures only the interactions between the features, resulting in smaller feature dimensions compared to simple concatenation. When a model is expanded to accommodate different data types, such as text and images, feature fusion offers a seamless method of merging features from various models created for

distinct data types. This approach is particularly advantageous for merging architectures and diverse input modalities. In the future, we aim to extend this work by working on multimodal datasets.

References

- [1] Sukhdeo S, Mishra S, Walmsley S. Human monkeypox: a comparison of the characteristics of the new epidemic to the endemic disease. *BMC Infect Dis.* 2022;22(1):928. <https://doi.org/10.1186/s12879-022-07900-7>.
- [2] Naemi A, Schmidt T, Mansourvar M, Naghavi-Behzad M, Ebrahimi A, Wiil UK. Machine learning techniques for mortality prediction in emergency departments: a systematic review. *BMJ Open.* Nov. 2021;11(11): e052663. <https://doi.org/10.1136/bmjopen-2021-052663>.
- [3] Ravi D, et al. Deep learning for health informatics. *IEEE J biomedical health Inf.* 2016;21(1):4–21.
- [4] Miotto R, Wang F, Wang S, Jiang X, Dudley JT. Deep learning for healthcare: review, opportunities and challenges. *Brief Bioinform.* 2018;19(6):1236–46
- [5] "Enhancing Monkeypox Detection through Data Analytics: A Comparative Study of Machine and Deep Learning Techniques", Kinjal A. Patel, Asadi Srinivasulu, Kuntesh Jani, Goddindla Sreenivasulu, research article in *Advances in Engineering and Intelligence Systems*, 2023
- [6] Krishnaraj Chadaga, Srikanth Prabhu, Niranjana Sampathila, Sumith Nireshwalya, Swathi S. Katta, Ru-San Tan, and U. Rajendra Acharya, "Application of Artificial Intelligence Techniques for Monkeypox: A Systematic Review", Systematic review, *Diagnostics*, 2023.
- [7] Katy Brown and Peter A. Leggat, "Human Monkeypox: Current State of Knowledge and Implications for the Future", tropical medicine and infectious disease, MDPI, 2016.
- [8] Sahin, V.H.; Oztel, I.; Yolcu Oztel, G. Human monkeypox classification from skin lesion images with deep pre-trained network using mobile application. *J. Med Syst.* 2022, 46, 79.
- [9] Irmak, M.C.; Aydin, T.; Yaanolu, M. Monkeypox Skin Lesion Detection with MobileNetV2 and VGGNet Models. In *Proceedings of the 2022 Medical Technologies Congress (TIPTEKNO)*, Antalya, Turkey, 31 October–2 November 2022; pp. 1–4
- [10] Shivangi Surati, Himani Trivedi, Bela Shrimali, Chintan Bhatt, and Carlos M. Travieso-González, "An Enhanced Diagnosis of Monkeypox Disease Using Deep Learning and a Novel Attention Model Senet on Diversified Dataset": Multimodal technologies and interaction, MDPI, 2023.
- [11] MonDiaL-CAD: Monkeypox Diagnosis via Selected Hybrid CNNs Unified with Feature Selection and Fusion Learning, Omneya Attallah, *Digital Health*, 2023.
- [12] Uysal F. Detection of Monkeypox Disease from Human Skin Images with a Hybrid Deep Learning Model. *Diagnostics.* 2023; 13(10):1772. <https://doi.org/10.3390/diagnostics13101772>.
- [13] Yolcu Oztel, G. Vision transformer and CNN-based skin lesion analysis: classification of monkeypox. *Multimed Tools Appl* **83**, 71909–71923 (2024). <https://doi.org/10.1007/s11042-024-19757-w>.
- [14] Muñoz-Saavedra, L.; Escobar-Linero, E.; Civit-Masot, J.; Luna-Perejón, F.; Civit, A.; Domínguez-Morales, M. A Robust Fusion of Convolutional Neural Networks for the Detection of Monkeypox Disease from Skin Images. *Sensors* 2023, 23, 7134. <https://doi.org/10.3390/s23167134>
- [15] Monkeypox Skin Images Dataset (MSID). Available online: <https://www.kaggle.com/datasets/dipuiucse/monkeypoxskinimagedataset> (accessed on 2 October 2022).
- [16] Kaggle. Monkeypox skin lesion dataset. Available: <https://www.kaggle.com/datasets/nafin59/monkeypox-skin-lesion-dataset>; 2022.
- [17] Sokolova, M.; Lapalme, G. A systematic analysis of performance measures for classification tasks. *Inf. Process. Manag.* 2009, 45, 427–437.
- [18] Sorayaie Azar et al. *BMC Infectious Diseases* (2023) 23:438 <https://doi.org/10.1186/s12879-023-084084>.
- [19] <https://www.ncbi.nlm.nih.gov/>, National Center for Biotechnology Information.
- [20] Haripriya, K.P., Hannah Inbarani, H. (2024). Performance Analysis of Various Feature Extraction Methods for Classification of Pox Virus Images. In: Jha, P.K., Tripathi, B., Natarajan, E., Sharma, H. (eds) *Proceedings of Congress on Control, Robotics, and Mechatronics. CRM 2023. Smart Innovation, Systems and Technologies*, vol 364. Springer, Singapore. https://doi.org/10.1007/978-981-99-5180-2_18
- [21] Abdelhamid, A.A.; El-Kenawy, E.-S.M.; Khodadadi, N.; Mirjalili, S.; Khafaga, D.S.; Alharbi, A.H.; Ibrahim, A.; Eid,

- M.M.; Saber, M. Classification of Monkeypox Images Based on Transfer Learning and the Al-Biruni Earth Radius Optimization Algorithm. *Mathematics* 2022, *10*, 3614. <https://doi.org/10.3390/math10193614>.
- [22] Tripathi, A., Bharti, K.K. & Ghosh, M. A fusion of binary grey wolf optimization algorithm with opposition and weighted positioning for feature selection. *Int. j. inf. technol.* 15, 4469–4479 (2023). <https://doi.org/10.1007/s41870-023-01481-7>.
- [23] Article on Remaining on alert: how monkeypox could spread back to animals from humans, <https://www.woah.org/en/article/remaining-on-alert-how-monkeypox-could-spread-back-to-animals-from-humans/>
- [24] Zhou, Tao & Ye, XinYu & Lu, HuiLing & Zheng, Xiaomin & Qiu, Shi & Liu, YunCan. (2022). Dense Convolutional Network and Its Application in Medical Image Analysis. *BioMed Research International*. 2022. 1-22. 10.1155/2022/2384830.
- [25] Seidaliyeva, Ulzhalgas & Akhmetov, Daryn & Ilipbayeva, Lyazzat & Matson, Eric. (2020). Real-Time and Accurate Drone Detection in a Video with a Static Background. *Sensors*. 20. 3856. 10.3390/s20143856.