Multimedia Tools and Applications (2024) 83:75061–75083
https://doi.org/10.1007/s11042-024-18416-4

MNPDenseNet: Automated Monkeypox Detection Using Multiple Nested Patch Division and Pretrained DenseNet201

Fahrettin Burak Demir[1] · Mehmet Baygin[2] · Ilknur Tuncer[3] · Prabal Datta Barua[4] · Sengul Dogan[5] · Turker Tuncer[5] · Chui Ping Ooi[6] · Edward J. Ciaccio[7] · U. Rajendra Acharya[8,9]

Received: 10 March 2023 / Revised: 22 November 2023 / Accepted: 22 January 2024 /
Published online: 15 February 2024

Abstract

Background Monkeypox is a viral disease caused by the monkeypox virus (MPV). A surge in monkeypox infection has been reported since early May 2022, and the outbreak has been classified as a global health emergency as the situation continues to worsen. Early and accurate detection of the disease is required to control its spread. Machine learning methods offer fast and accurate detection of COVID-19 from chest X-rays, and chest computed tomography (CT) images. Likewise, computer vision techniques can automatically detect monkeypoxes from digital images, videos, and other inputs.

Objectives In this paper, we propose an automated monkeypox detection model as the first step toward controlling its global spread.

Materials and method A new dataset comprising 910 open-source images classified into five categories (healthy, monkeypox, chickenpox, smallpox, and zoster zona) was created. A new deep feature engineering architecture was proposed, which contained the following components: (i) multiple nested patch division, (ii) deep feature extraction, (iii) multiple feature selection by deploying neighborhood component analysis (NCA), Chi2, and ReliefF selectors, (iv) classification using SVM with 10-fold cross-validation, (v) voted results generation by deploying iterative hard majority voting (IHMV) and (vi) selection of the best vector by a greedy algorithm.

Results Our proposal attained a 91.87% classification accuracy on the collected dataset. This is the best result of our presented framework, which was automatically selected from 70 generated results.

Conclusions The computed classification results and findings demonstrated that monkeypox could be successfully detected using our proposed automated model.

Keywords MNPDenseNet · Monkeypox detection · Image processing · Biomedical engineering

Extended author information available on the last page of the article

1 Introduction

Poxviruses are complex viruses with large double-stranded deoxyribose nucleic acid (DNA), and they cause disease in humans and many other types of animals [1]. Poxviruses are from the family Poxviridae, where the latter is subdivided into two different subfamilies: Chondropoxvirinae (poxviruses of vertebrates) and Entomopoxvirinae (poxviruses of insects) [2, 3]. In the subfamily Chondropoxvirinae, the orthopoxviruses genus that has infected humans include variola (smallpox) [4], monkeypox, and cowpox [1]. Smallpox was eradicated globally in 1979, while human cases of monkeypox and cowpox are still being reported to this day.

The rising number of monkeypox cases reported in countries where the disease is not endemic has alerted the global medical community. The disease is now considered of global public health importance as it spreads worldwide [5, 6]. The symptoms of monkeypox often include fever and rash, similar to smallpox [7]. These similarities are important for our diagnostic study.

Monkeypox, as the name implies, was a viral disease known at one time only to occur in monkeys [5, 6]. However, the disease was first identified in humans in 1970 in the Democratic Republic of Congo [8, 9]. Since then, monkeypox outbreaks have mostly been confined to countries in West and Central Africa, such as Nigeria, the Central African Republic, Liberia, Sierra Leone, and Cameroon. The first outbreak of monkeypox outside of Africa was in the USA in 2003. In 2022, multiple cases of monkeypox have been identified in several regions of the world, including non-endemic countries.

The monkeypox virus is spread through physical contact with the blood, bodily fluid, or lesion material from infected individuals or contaminated materials. Hence, maintaining good personal hygiene such as frequent washing of hands with soap and water, regular washing of clothes and social distancing can prevent disease transmission. Besides human-to-human transmission, animal-to-human transmission may also occur by a bite or scratch from an infected animal, eating the uncooked meat of an infected animal, or handling infected animals [10].

There are no proven treatments or vaccines for monkeypox infection but the effectiveness of two doses of the Jynneos vaccine is about 87% [11]. However, studies on the smallpox vaccine have shown it to provide some protection against other orthopox virus varieties, including monkeypox. In addition, approximately 85% of individuals previously vaccinated against smallpox were found to be resistant to the monkeypox virus [12, 13].

Computer vision models are very useful due to their high classification performances [14, 15]. Especially, patch-based models, for instance, vision transformers (ViT) [16], multilayer perceptron mixer (MLP-mixer) [17], and ConvMixer [18] have attained high classification performances. Several biomedical image classification models have been proposed which have significant advantages. In the COVID-19 pandemic, variable computer vision-based COVID-19 detection models have been proposed. We have proposed a new framework to detect monkeypox disorder using images in this research. Moreover, our main objective is to handle the time complexity problem of the fixed-sized patches-based feature extraction problems, since the optimal size of the used patches is a nondeterministic polynomial problem. Thus, we have used nested patches, as discussed following.

1.1 Literature Review

Skin image classification is one of the most popular topics in machine learning. Especially advances in artificial intelligence and image-based diagnosis and detection approaches offer important solutions in the field of healthcare. Table 1 summarizes the studies on machine learning-based classification of poxviruses.

Ahsan et al. [19] collected the data for their research from Google. They performed two tests. The first test compares monkeypox and chickenpox. In the second test, monkeypox and others were compared. In this test, monkeypox data was subjected to data augmentation. Sitaula and Shahi [20], tested the data collected by Ahsan et al. [19]. Sitaula and Shahi combined 13 pre-trained deep network architectures using majority voting and obtained an ensemble model. Islam et al. [22] created their dataset consisting of a total of 804 images and 6 classes. In the test process, they applied the 5-fold CV strategy and achieved an average classification success of 83%. Sahin et al. [23] used an open-access dataset and applied binary classification. Abdelhamid et al. [25] performed binary classification with this method and achieved a classification success rate of over 98%. Bala et al. [26] created their own dataset with four classes. They tested this dataset with the hold-out CV method and obtained 98.91% accuracy in the test process with the augmented method. Ozsahin et al. [27] performed binary classification with a self-designed CNN. Yasmin et al. [28] classified monkeypox and others using an open-access dataset. In the research, they used their self-designed CNN method and increased the data with data augmentation method. As a result of this process, it achieved 100% classification success.

1.2 Motivation and our model

Machine learning and deep learning methods are useful in assisting researchers and medical professionals in the diagnosis and detection of COVID-19 cases [29–31]. Hence, we believe these methods could similarly be applied to monkeypox detection. In this paper, we propose to use machine learning for fast and accurate detection of monkeypoxes from imagery.

The dataset contains images of healthy skin and images from 4 different viral diseases that cause rash and blisters (monkeypox, chickenpox, smallpox, and zoster zona) obtained from the public database. In this work, we have proposed a novel deep feature engineering model to discern these classes automatically. Our model uses multiple nested patches of different initial sizes (14 × 14, 16 × 16, 28 × 28, 32 × 32, 56 × 56, and 112 × 112). The two deep learning features used (these are obtained from pretrained DenseNet201) are global average pooling and fully connected layers. Using six patch types and two feature extraction functions, 12 (= 6 × 2) feature vectors of various sizes are extracted. Applying three selectors and SVM, 36 (= 12 × 3) predicted vectors are used as input of the IHMV. The IHMV then generates 34 (= 36 - 2) voted predicted vectors and the predicted vector with the highest accuracy.

1.3 Contributions and novelties

The novelty and contributions of this research are given below.

- Use of a new image dataset of four viral diseases that cause rash and/or blisters (monkeypox, chickenpox, smallpox, and zoster zona).
- A novel monkeypox image classification model is developed using a deep feature engineering model.

Table 1 Poxvirus classification with machine learning using skin images

Author(s) and Year	Classes	Source of Dataset	Method	Validation	Result
Ahsan et al., 2022 [19]	Monkeypox, Chickenpox, Measles, Normal	Google search	Data augmentation, VGG16-based transfer learning	80:20	Test-1: 83.0 Test-2: 78.0
Sitaula and Shahi, 2022 [20]	Monkeypox, Chickenpox, Measles, Normal	Google search [19, 21]	Ensemble pretrained deep learning models, majority voting	5-fold CV	87.13
Islam et al., 2022 [22]	Monkeypox, Chickenpox, Smallpox, Cowpox, Measles, Normal	Google search	Data augmentation, ShuffleNet-V2, Ensemble CNN, Majority voting	5-fold CV	83.0
Sahin et al., 2023 [23]	Monkeypox, Non-monkeypox	Kaggle [24]	Data augmentation, Image preprocessing, and Transfer learning using six pretrained CNN	70:10:20	91.11
Abdelhamid et al., 2022 [25]	Monkeypox, Normal	Kaggle [26]	Data augmentation, GoogleNet-based deep feature extraction, meta-heuristic optimization algorithm-based feature selection, Neural network	---	98.8
Bala et al., 2023 [26]	Monkeypox, Chickenpox, Measles, Normal	Kaggle [26]	Data augmentation, Classical machine learning classifier and deep learning models, custom-designed CNN,	80:20	Test-1: 93.19 Test-2: 98.91
Ozsahin et al., 2023 [27]	Monkeypox, Chickenpox	Kaggle [24]	Custom designed CNN	---	99.60
Yasmin et al., 2023 [28]	Monkeypox, Non-monkeypox	Kaggle [24]	Data augmentation, InceptionV3 based custom designed CNN (PoxNet22)	80:20	100

- A new multiple-patch division model is being proposed.
- We have also proposed a self-organized framework that automatically selects the best-performing model.

2 Materials and Methods

2.1 Dataset

Nine hundred and ten skin images were collected from the web and classified into five categories. The categories are (i) monkeypox, (ii) chickenpox, (iii) smallpox, (iv) healthy, and (v) zoster zona. Skin images of smallpox, chicken pox, and zoster zona were included because these diseases cause similar skin afflictions (rash and/or blisters) to monkeypox. The collected images were stored as JPG and PNG and were of various sizes. The number of images in each category is tabulated in Table 2.

Representative skin images from each category are shown in Fig. 1.

2.2 Proposed image classification model

In this work, we have proposed deep feature engineering, with the feature extraction capability of the patch-based models. For a lightweight model, DenseNet201 [32], a popular pretrained CNN [33], was deployed. Global average pooling and fully connected layers of this CNN were used to extract deep features. Six nested patch divisions were utilized to evaluate the features and to find the most appropriate patch size. Three commonly used feature selection functions were used: NCA [34], Chi2 [35], and ReliefF [36]. From this, 36 transfer learning-based feature engineering models were obtained and classified by deploying an SVM [37, 38] classifier. IHMV was employed to obtain voted results, with the best result chosen using a greedy algorithm. Since this model is implemented with variable-sized nested patch divisions and DenseNet201 for image classification, we have named it MNPDenseNet, where the MNP stands for multiple nested patches. The block diagram of the proposed MNPDenseNet model is depicted in Fig. 2.

The schematic in Fig. 2 demonstrates that our proposed MNPDenseNet is a self-organized architecture, and it can automatically select the best result from the generated 70 predicted vectors. As shown in Fig. 2, an end-to-end training approach was not used in this research. Instead, the DenseNet-201 architecture was chosen as the feature extractor. In this way, the network was not retrained and the weights of the network were not computed. In the model, the image was first divided into 6 patches and two separate feature vectors were obtained for each patch using 2 layers of the DenseNet-201 architecture. In this way, 12 feature vectors were obtained. The ensemble feature selection approach is used in the model. We favored 3 different feature

Table 2 Number of images in each category of the skin dataset

Number	Category	Number of images
1	Monkeypox	217
2	Chickenpox	127
3	Smallpox	151
4	Healthy	195
5	Zoster Zona	220
Total		910

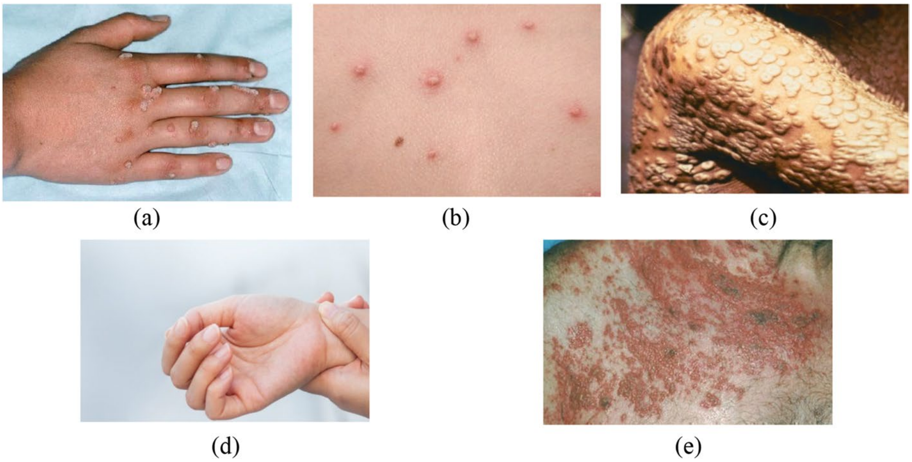
(a) (b) (c) (d) (e)

Fig. 1 Representative images of the collected dataset: (**a**) Monkeypox, (**b**) Chickenpox, (**c**), Smallpox (**d**) Healthy, (**e**) Zoster zona

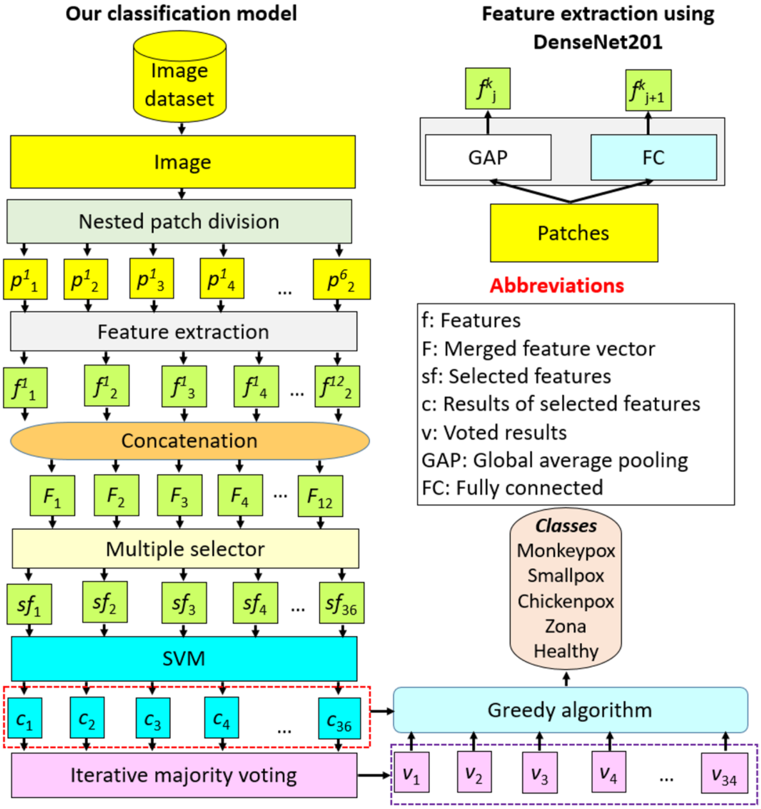
Our classification model
Image dataset
Image
Nested patch division
p^1_1 p^1_2 p^1_3 p^1_4 ... p^6_2
Feature extraction
f^1_1 f^1_2 f^1_3 f^1_4 ... f^{12}_2
Concatenation
F_1 F_2 F_3 F_4 ... F_{12}
Multiple selector
sf_1 sf_2 sf_3 sf_4 ... sf_{36}
SVM
c_1 c_2 c_3 c_4 ... c_{36}
Iterative majority voting
Feature extraction using DenseNet201
f^k_j
f^k_{j+1}
GAP
FC
Patches
Abbreviations
f: Features
F: Merged feature vector
sf: Selected features
c: Results of selected features
v: Voted results
GAP: Global average pooling
FC: Fully connected
Classes
Monkeypox
Smallpox
Chickenpox
Zona
Healthy
Greedy algorithm
v_1 v_2 v_3 v_4 ... v_{34}

Fig. 2 Block diagram of proposed MNPDenseNet model with input imaged dataset

selectors. These are NCA, Chi2, and ReliefF, respectively. A total of 36 (=3 feature selectors x 12 feature vectors) selected feature vectors were obtained from each feature vector using these feature selectors. In the classification phase of the model, the 36 selected feature vectors were separately classified using the SVM algorithm, resulting in 36 classification results. These results are the prediction vectors. In the final stage of the model, the results of these prediction vectors were combined using the IHMV algorithm and the best classification result was obtained using the Greedy algorithm. The MNPDenseNet architecture contains six layers. which are:

(i) preprocessing,
(ii) feature extraction,
(iii) feature selection,
(iv) classification,
(v) majority voting, and
(vi) selection of best result.

The details of these six layers are explained below.

Preprocessing: The multiple nested patch (MNP) division was applied to the original image. The original image was first converted to a 224 × 224 dimension before six nested patch divisions (14 × 14, 16 × 16, 28 × 28, 32 × 32, 56 × 56, and 112 × 112) were used to create six patch categories. These patch categories were then entered into the pretrained DenseNet201. The pseudocode of our MNP-based preprocessing layer is depicted in Algorithm 1.

```
Input: Skin image ($SI$)
Output: Patch categories ($p$)
00: Load $SI$.
01: Resize $SI$ to 224 × 224 sized images.
02: $pa = \{14,16,28,32,56,112\}$ // $pa$ defines patch array.
03: $center = 112$;
04: for i=1 to 6 do
05:     $inc = \frac{pa(i)}{2}$; // Herein, $inc$ defines increasing value.
06:     $cnt = 1$; // $cnt$ defines counter.
07:     for j=1 to $\frac{224}{pa(i)}$ do
08:         $p_{cnt}^{i} = SI(center - inc + 1{:}\, center + inc, center - inc + 1{:}\, center + inc)$
// In Line 08, nested patch division has been given, and $p$ defines the patch category.
09:         $cnt = cnt + 1$;
10:     end for j
11: end for i
```

Algorithm 1. The proposed MNP-based preprocessing method

By using these six sizes (initial sizes are 14 × 14, 16 × 16, 28 × 28, 32 × 32, 56 × 56, and 112 × 112), six patch categories resulted, which contain 16, 14, 8, 7, 4, and 2 non-fixed sized patches, respectively. By deploying these patches, feature extraction based on pretrained DenseNet201 has been processed. In this section, the first step of the proposed architecture has been given.

Step 1: Apply Algorithm 1 to images for creating patches

In this phase, facial areas of images are segmented and the cropped/segmented images are converted to grayscale images. The obtained grayscale images are resized into 224 × 224-dimensional images.

Feature extraction: We used two layers of a pretrained network to extract deep features, and incorporated a pretrained DenseNet201 as a feature extractor. The DenseNet201 was trained on ImageNet1K [33]. We used DenseNet201 feed-forwarded, and we did not use the Softmax layer. Outputs of two layers have been utilized as feature vectors: global average pooling (GAP) and fully connected (FC) layers. By using the GAP layer, 1,920 features have been extracted from the GAP layer, and 1,000 features have been extracted from the FC layer. By using two layers, two deep feature extractors have been obtained from pretrained DenseNet201. The GAP layer calculates the average value for each channel, summarizing the features of the image, and capturing general patterns and characteristics in the images, such as the type and prevalence of objects or patterns. These features are then processed by the FC layer, which is responsible for further refinement and classification, with each of the 1,000 features being essential for accurate image classification tasks, representing specific classes or object types. These features play a fundamental role in our deep learning model, enabling it to learn from the data, recognize important patterns, and accurately classify images. They serve as critical information for the success of image classification tasks. The graphical outline of the DenseNet201-based feature extractors is summarized in Fig. 3.

Step 2: Extract features from the created patches in the preprocessing layer by deploying the FC layer of the pretrained DenseNet201.

$$c_j^k = dfc\left(p_j^k\right), k \in \{1, 2, \ldots, 6\}, j \in \{1, 2, \ldots, n\}, n \in \{16, 14, 8, 7, 4, 2\} \quad (1)$$

In this step, features are extracted from the created patches using the Fully Connected (FC) layer of the pretrained DenseNet201. This layer, represented by c, acts as a feature extractor and is defined as the function $dfc(.)$. It generates 1,000 features from each patch. The notation c_j^k represents the feature vector extracted from the FC layer for the patch p_j^k, where k varies from 1 to 6, j varies from 1 to n and n can take on values 16, 14, 8, 7, 4, or 2.

Step 3: Apply the GAP layer of the pretrained DenseNet201 to the generated patches.

$$g_j^k = dgap\left(p_j^k\right) \quad (2)$$

In this step, the generated patches are further processed by applying the Global Average Pooling (GAP) layer of the pretrained DenseNet201. The features extracted through this process are denoted as g_j^k. Each g_j^k feature vector has a length of 1,920, and $dgap(.)$ represents the function used for GAP-based feature extraction.

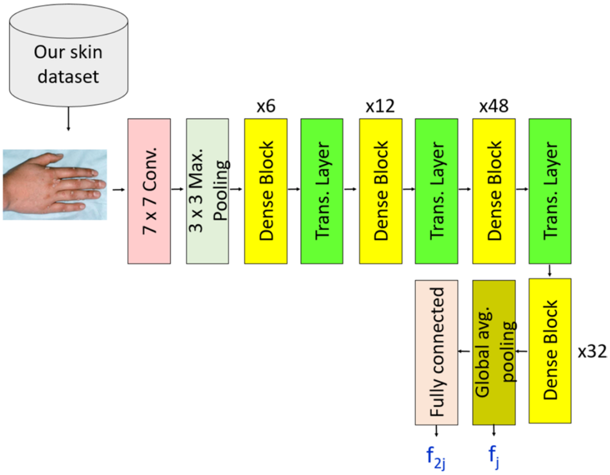

Fig. 3 Outline of the DenseNet201-based feature extractors using GAP and FC layers. Herein, transition layers have been denoted using Trans Layer and maximum pooling is demonstrated using Max Pooling

Step 4: Create features using the concatenation function.

$$F^{2k-1} = concat\left(c_1^k, c_2^k, \dots, c_n^k\right) \tag{3}$$

$$F^{2k} = concat\left(f_1^k, f_2^k, \dots, f_n^k\right) \tag{4}$$

In the final step, features are created by concatenating the feature vectors obtained from the FC layer and the GAP layer. There are two sets of features created: F^{2k-1} and F^{2k}. These merged feature vectors are represented by F. The function $concat(.)$ is used for concatenation. Altogether, a total of 12 feature vectors are generated through this process.

Feature selection: We used three common feature selection functions: neighborhood component analysis (NCA), Chi2, and ReliefF feature selectors to investigate the feature selection abilities. The brief explanations of these selectors are given below.
NCA: It is a distance-based feature selection function. Therefore, it is named the feature selection version of the kNN classifier. It uses L1-norm distance (Manhattan distance) and stochastic gradient descent optimizer to compute the weights of the features, and these features are non-negative features. By using the computed weights, features have been qualified [34].
Chi2: It is a statistical-based feature selector, and indices of the sorted weights have been used for a Chi2 statistical function. The Chi2 selector is one of the fastest feature selectors described in the literature. Thus, we used the Chi2 feature selection function in our architecture [35].
ReliefF is a developed version of the Relief feature selection function and generates weights like NCA, but ReliefF generates both positive and negative weights. Negative weighted features are redundant features, according to ReliefF. Therefore, the best features have been selected by sorting the generated weights [39].

We employed these functions in the feature selection phase, where feature vectors were generated using DenseNet-201. The primary goal was to reduce the dimensionality of the feature vectors, subsequently lowering computational complexity.

- NCA, Chi2, and ReliefF for Feature Selection: NCA, Chi2, and ReliefF were applied to the feature vectors generated by DenseNet-201. These feature selection functions assess the relevance of individual features, allowing us to identify the most informative features in the dataset.
- Dimensionality Reduction: By utilizing NCA, Chi2, and ReliefF, we aimed to reduce the dimensionality of the feature vectors while preserving the most discriminative attributes. This dimensionality reduction enhances computational efficiency and reduces the risk of overfitting.

These feature selection functions allow working with more manageable feature vectors without compromising the quality of the feature vectors. Therefore, in our research, we have reduced the size of the feature vectors in a way that does not negatively affect the classification success and aimed for high classification success/low computational complexity.

Step 5: Apply NCA, Chi2, and ReliefF feature selection functions to the generated 12 feature vectors.

$$ind^{3h-2} = \gamma\left(F^h\right), h \in \{1,2,\ldots,12\} \tag{5}$$

$$ind^{3h-1} = \chi\left(F^h\right) \tag{6}$$

$$ind^{3h} = \vartheta\left(F^h\right) \tag{7}$$

In this step, three different feature selection functions are applied to the 12 feature vectors generated in the previous steps. These functions are represented as $\gamma(.)$, $\chi(.)$ and $\vartheta(.)$ and correspond to NCA, Chi2, and ReliefF feature selection methods. The outcome of this step is the sorted indexes of the features, denoted as ind^{3h-2}, ind^{3h-1} and ind^{3h} for each of the 12 feature vectors F^h (where h varies from 1 to 12).

Step 6: Choose the top 512 features (like ternary pattern) using the generated indexes.

$$sf^{3h-2}(i,j) = F^h\left(i, ind^{3h-2}(j)\right), i \in \{1,2,\ldots,dim\}, j \in \{1,2,\ldots,512\} \tag{8}$$

$$sf^{3h-1}(i,j) = F^h\left(i, ind^{3h-1}(j)\right) \tag{9}$$

$$sf^{3h}(i,j) = F^h\left(i, ind^{3h}(j)\right) \tag{10}$$

In this final step, the top 512 features are selected from each feature vector using the generated indexes. The selected features are represented as sf^{3h-2}, sf^{3h-1}, and sf^{3h} for each of the 12 feature vectors. The notation sf signifies the selected features, and there are 36 selected features in total, each with a length of 512. dim represents the number of images, which, in this dataset, is 910 images.

Classification: We used a shallow classifier to obtain classification results for the 36 generated feature vectors. The shallow classifier utilized is SVM, and we selected it in the MATLAB Classification Learner toolbox. In the MATLAB Classification Learner tool, there are 30 shallow classifiers. We used this toolbox to select the most appropriate classifier and employed the classifiers with default settings. The best-resulting classifier is Cubic SVM [40]. Hence, we classified the generated 36 feature vectors by deploying Cubic SVM with ten-fold cross-validation. In this layer, 36 predicted vectors have been generated.

Step 7: Generated predicted vectors were created by deploying a SVM classifier.

$$prv^t = \xi\left(sf^t, y\right), t \in \{1, 2, \dots, 36\} \tag{11}$$

Herein, prv defines predicted vectors, $\xi(.,.)$ is a Cubic SVM function, and y represents real labels. The attributes of the used SVM classifier are:

Kernel: Polynomial,
Kernel order: 3,
Kernel scale: Automatic,
Box constraint: 1,
Coding: One-vs-all,
Validation: 10-fold CV.

Iterative Hard Majority Voting: The IHMV was proposed by Dogan et al. [41] in 2021 to calculate more voted results. In the IHMV, we use the mode function to obtain voted results. Firstly, the generated predicted vectors in the classification layer have been utilized as input for the IHMV algorithm, and these results are sorted by their classification accuracies. Then, a loop has been created, and the loop range is 3 to 36. Therefore, 34 (= 36 - 3 + 1) voted results have been created in this layer. To better explain this function, the pseudocode of the IHMV is shown in Algorithm 2.

```
Input: Predicted vectors ($prv$)
Output: Voted vectors ($vv$)

Calculate the classification accuracy of all predicted vectors.
Sort predicted vectors by their accuracies.
for q=3 to 36 do
    for i=1 to $dim$ do
        for j=1 to q do
            $array(j) = prv^{idx(j)}(i)$;
// Array creating for voting. $idx$ defines sorted indexes
        end for j
        $vv^{q-2}(i) = \varrho(array)$;
// where $\varrho(.)$ is an array function. Herein, mode-based majority voting is processed.
    end for i
end for q
```

Algorithm 2. IHMV algorithm

Step 8: Created 34 voted vectors by deploying the IHMV function

Selection: In the classification and IHMV layers, 70 (=36+34) predicted vectors were created. The best-resulted vector has been selected in this layer by deploying the greedy method. The greedy algorithm is a commonly used selection method and it has been implemented in metaheuristic optimization techniques to select the best solution [42]. By using this layer in the developed model, maximum performance is achieved without the need for manual selection. Herein, 70 predicted vectors have been generated and the best accurate predicted vector was selected by deploying the greedy algorithm. Using this layer, our proposed MNPDenseNet architecture became a self-organized architecture. The last two steps of our proposal are given in this layer, i.e.:

Step 9: Calculate classification accuracies of the generated 70 predicted vectors.

Step 10: Select the highest accurate vector as a final result.

The transition of the MNPDenseNet model is summarized in Table 3.

3 Results

3.1 Experimental Setup

The proposed MNPDenseNet model was programmed using MATLAB ver. 2020a on a simple configured PC (16 GB main memory, intel i7 7700 processing unit, Windows 11) without the need for graphical or tensor processing units. Seven performance metrics - accuracy, overall recall (OR), overall precision (OP), overall F1-score (OF1), geometric mean (GM), cohen's kappa (CK) and Matthews correlation coefficient (MCC) [43, 44] - were employed to evaluate the proposed model.

3.2 Results

We have proposed a framework, and this framework generates 70 results. The best result is the 38th result, and our proposed MNPDenseNet selected it. Therefore, the 38th (2nd voted vector) result was generated, voting the top four results (28th – 4^2 patches (56 × 56 sized patches) + GAP layer + NCA –, 10th – 14^2 patches (16 × 16 sized patches) + GAP layer + NCA –, 22nd – 7^2 (32 × 32 sized patches) patches + GAP layer + NCA –, and 34th – 2^2 (112 × 112 sized patches) patches + GAP layer + NCA –). To generate these top four feature vectors, the used feature extraction function GAP layer of the DenseNet201 and the used selector is NCA. These top four results have been fused and the best-resulted output has been generated. The confusion matrix of the best result is also given in Fig. 4 to calculate classification performances.

The classification performances of the proposed MNPDenseNet are summarized in Table 4. As seen in Table 4, the proposed MNPDenseNet model attained over 91% for all performance metrics.

The class-wise recall, precision, and F1-score for the different classes are tabulated in Table 5. The most accurate class was Healthy, with 98.46% recall for this category, while the least accurate was Chickenpox with 86.61%. It has to be noted that the Chickenpox class had the smallest number of skin images (127).

Table 3 Steps involved in MNPDenseNet model

Layers	Process	Size/output
Preprocessing	Image resizing	224×224
	MNP	Number of patches Categories P1: 16, P2: 14, P3: 8, P4: 7, P5: 4, P6: 2
Feature extraction	Deep feature generation using FC and GAP layer of the pretrained DenseNet201	Length of features: F1: 16000, F2: 30720 (extracted from P1) F3: 14000, F4: 26880 (extracted from P2) F5: 8000, F6: 15360 (extracted from P3) F7: 7000, F8: 13440 (extracted from P4) F9: 4000, F10: 7680 (extracted from P5) F11: 2000, F12: 3840 (extracted from P6)
Feature selection	NCA, Chi2, and ReliefF selectors were used to choose features.	36 feature vectors (sf) have been selected. The length of each sf is 512. sf^{3h-2} are generated using NCA,sf^{3h-1} are generated using Chi2 and sf^{3h} are generated using ReliefF. Herein, $h \in \{1, 2, \ldots, 12\}$.
Classification	Cubic SVM with 10-fold CV	36 predicted vectors have been generated
Voting	IHMV	Loop range: [3, 36] Voting function: Mode Weighting: None Number of voted vectors: 34
Selection of the best result	Greedy algorithm	Selection of the best-predicted vector by classification accuracies of the 70 predicted vectors.

3.3 Time Complexity

Our model is a lightweight approach compared to classic CNN architectures. The proposed model uses deep feature extraction instead of the end-to-end training phase in classical CNNs. Therefore, there is no need to calculate the weights of the network. In this context, the time complexity of the developed model is calculated in Table 6.

4 Discussions

In this study, a new deep-feature engineering architecture was developed to automatically detect monkeypox from skin images. A new large dataset of skin images (from healthy individuals and those with monkeypox, smallpox, chickenpox, and shingles) was created from images obtained from the publicly available dataset and supplemented with relevant web images. The proposed MNPDenseNet model generated 70 results – the number of results denotes that the proposed MNPDenseNet tested all configurations to obtain the most accurate combination among the 70 result, and selected the best validation prediction vector for best classification result. The classification accuracies of the generated 70 results are depicted in Fig. 5.

As seen in Fig. 5, the most accurate predicted vector is the 38th predicted vector, generated using IHMV. Thus, the 38th predicted vector is a voted vector. This vector was generated using the most accurate four feature vectors (in the first 36 predicted vectors). These vectors are 28, 10, 22, and 34, and they achieved 91.65%, 91.32%, 91.32%, and 91.32%

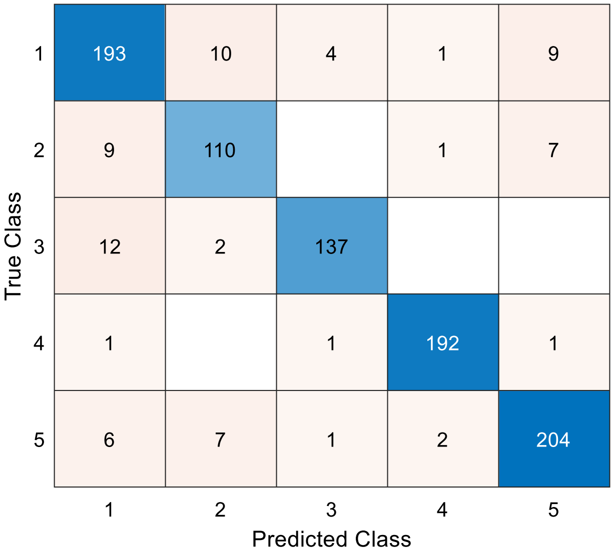

Fig. 4 Confusion matrix of our proposal. 1: Monkeypox, 2: Chickenpox, 3: Smallpox, 4: Healthy, 5: Zoster zona

Table 4 Performance metrics (%) of the proposed MNPDenseNet model

Performance metrics	Results (%)
Accuracy	91.87
Overall Recall	91.49
Overall Precision	91.73
F1-score	91.60
Geometric mean	91.41
Cohen’s Kappa	89.72
MCC	89.56

accuracies, respectively. These results belong to $sf_{28}, sf_{10}, sf_{22}$ and sf_{34}. All of these features were generated using the GAP layer and NCA selectors. GAP is more effective in this case than the FC layer, and the top four features were created using the NCA selector. Moreover, these four vectors ($sf_{28}, sf_{10}, sf_{22}, sf_{34}$) were generated using 56 × 56, 16 × 16, 32 × 32, and 112 × 112 sized nested patches, respectively. By voting these results, the best result is achieved. Moreover, the least accurate predicted vector is the 2nd predicted vector - generated using sf_2 and using FC layer with 14 × 14 sized nested patch and Chi2 selector. Still, this vector resulted in an 81.98% classification accuracy.

4.1 Ablation of the proposed model

To discuss the effect of the size of the patches, average classification accuracies of the generated predicted vectors are calculated per the used sizes of the patches. Comparison results according to patch sizes are given in Fig. 6.

Figure 6 shows that the best patch division method is 56 × 56, with an average classification accuracy of 88.15%. Moreover, we have used two feature extractors and three

Table 5 Class-wise recall, precision, and F1-score of the proposed MNPDenseNet model

Class	Recall (%)	Precision (%)	F1-score (%)
Monkeypox	88.94	87.33	88.13
Chickenpox	86.61	85.27	85.94
Smallpox	90.73	95.80	93.20
Healthy	98.46	97.96	98.21
Zoster zona	92.73	92.31	92.52

feature selectors to create the feature vectors. The average classification accuracies of the feature vector creation methods are given in Fig. 7.

As shown in Fig. 7, the best feature creation method is GAP + NCA, and the average classification accuracy of this feature creation model is equal to 91.23%. Already, the best feature vector (sf_{28}) was created using 56 × 56 sized patches, a GAP feature extractor, and NCA feature selector. Figure 5 confirms the high classification result of the 28th selected feature. Furthermore, we can say that the best feature extractor and feature selectors are the GAP layer of the DenseNet201 and the NCA feature selector, respectively. The effect of using patches in the ablation test phase of the model was observed. For this, patch and non-patch-based architectures were compared. The result of this comparison is given in Fig. 8.

As shown in Fig. 8, the patch-based method achieved about 20% higher classification performance. Another test conducted within the scope of ablation studies is the measurement of the IHMV effect. In this test process, IHMV vs. Non-IHMV status was compared. The results of the test process are given in Fig. 9.

As shown in Fig. 9, the IHMV-based classification results are higher than the classification results without IHMV. The results given in Figs. 6, 7, 8 and 9 prove the effectiveness of the methods used in the model. With these methods, our MNPDenseNet model achieved 91.87% classification accuracy.

Table 6 Time complexity of the proposed model

Phase	Time Complexity
Image Resizing	$O(n)$
Nested Patch Division	$O(nd)$
Feature Extraction with DenseNet201	$O(td)$
Feature Selection with NCA, Chi2 and ReliefF	$O(s)$
Classification with SVM	$O(fd)$
IHMV	$O(id)$
Greedy Algorithm	$O(l)$
Total	$O(n + nd + td + s + fd + id + l)$

*n: number of images, d: number of patches, t: time complexity of DenseNet201, s: time complexities of feature selectors, f: number of features, i: number of iterations, l: number of predicted vectors.

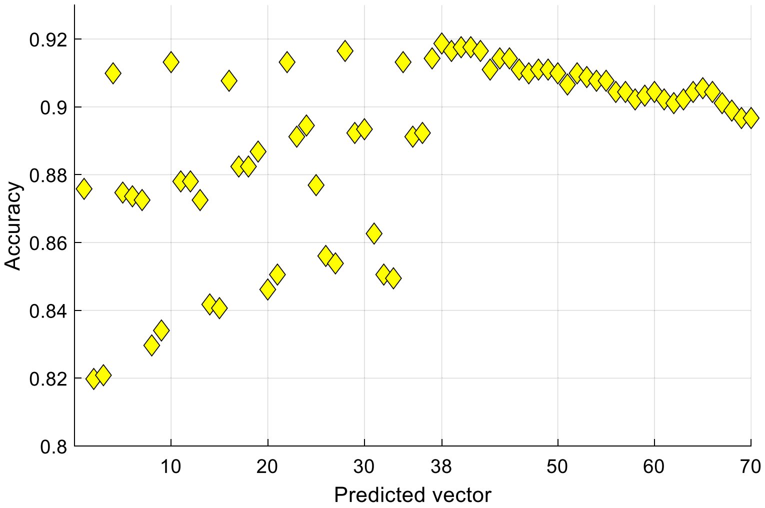

Fig. 5 Plot of calculated classification accuracies per calculated predicted vector

4.2 Comparative Results

To obtain comparative results, pretrained AlexNet, MobileNetv2, DarkNet53, and ResNet50 were used. The comparative results from these pretrained networks and our MNPDenseNet model are tabulated in Table 7.

In MNPAlexNet, fc6 and fc7 layers were used as feature extractors. the remaining CNN, GAP and FC layers were utilized as feature extractors. It can be observed that the most appropriate pretrained CNN for our architecture to classify monkeypox images is DenseNet201. The least accurate model is MNPAlexNet since AlexNet uses fewer layers than the others. As presented in the previous sections, our model uses SVM for the classification phase. In this research, six different classifiers are considered in the test phase. These are Decision Tree (DT), Linear Discriminant (LD), Naive Bayes (NB), k-Nearest Neighbor (kNN), Support Vector Machine (SVM) and Neural Network (NN). The performance comparison of these classifiers is given in Fig. 10.

As shown in Fig. 10, the highest classification accuracy was achieved with the SVM algorithm and the lowest classification performance was achieved with DT. To prove the superiority of our model, we compared it with similar studies in the literature. Ahsan et al. [45] conducted a similar study using deep learning methods on monkeypox detection. They used a publicly available dataset containing 171 images with four classes [21, 46]: 1: Monkeypox, 2: Chickenpox, 3: Measles, 4: Healthy. Their deep learning-based model classified monkeypox [45] into two cases of binary classification (Case 1: Monkeypox vs. Chickenpox and Case 2: Monkeypox vs. Others). We applied our MNP-DenseNet model to this dataset and the comparative results are summarized in Table 8.

Table 8 demonstrated that our proposal attained higher classification accuracies than a VGG16-based image classification model used by Ahsan et al., and our model attained over 10% classification accuracy compared with the state-of-the-art method.

There are highly effective and efficient CNN models in the literature. However, the traditional approach involves end-to-end training of these CNN models with new data. Although these models produce high classification results, it is a time-consuming

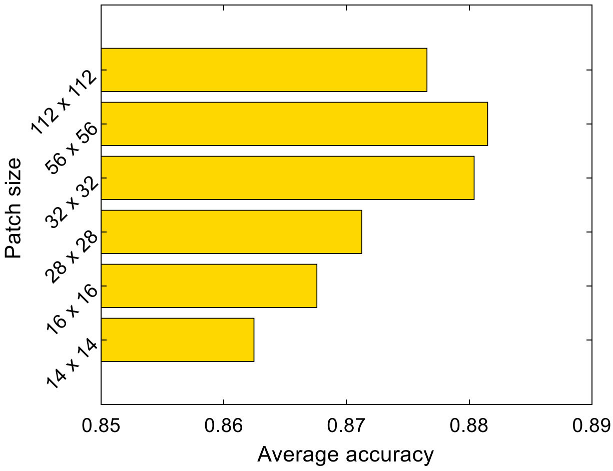

Fig. 6 Average classification accuracies according to the used patch divisions

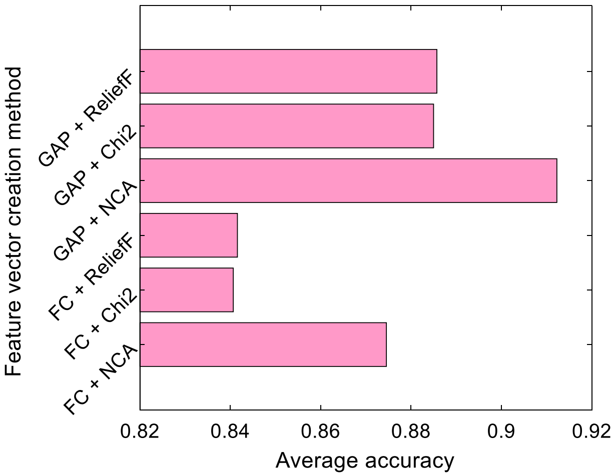

Fig. 7 Average classification accuracies according to the used feature creation method

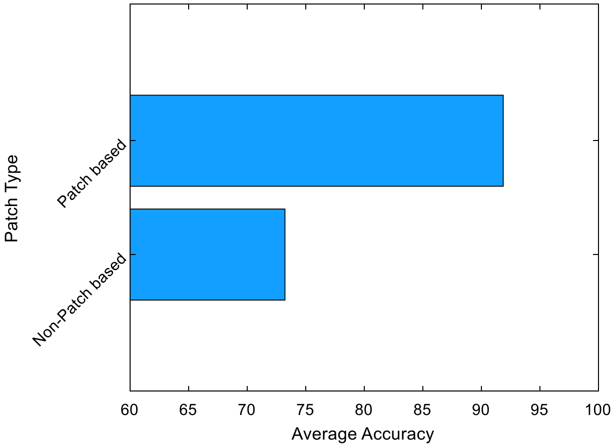

Fig. 8 Average classification accuracies according to the patch and non-patch-based method

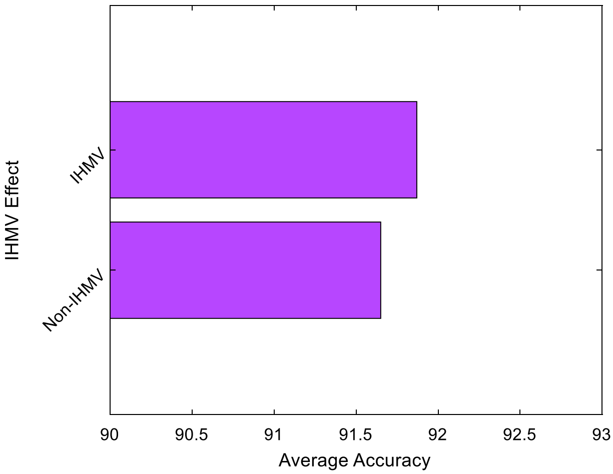

Fig. 9 Average classification accuracies according to the IHMV and non-IHMV-based method

Table 7 Comparative results (%) obtained with other models

Method	Accuracy(%)	Overall Recall(%)	Overall Precision(%)	F1-score(%)	Geometric mean(%)
MNPAlexNet	87.36	86.45	86.88	86.64	86.16
MNPMobileNet	88.24	87.68	87.76	87.71	87.49
MNPDarkNet	89.01	88.38	88.78	88.48	88.27
MNPResNet	90.22	89.57	89.93	89.71	89.42
MNPDenseNet	91.87	91.49	91.73	91.60	91.41

process. Moreover, these models require a lot of data for high classification success. The model proposed in this research is designed to address these and similar challenges. The technical advantages of the proposed model are given below:

- Reduced Time Complexity: Our model has lower time complexity compared to end-to-end CNN models. It has an architecture that is prone to work with limited computational resources.
- Data Efficiency: Our model can handle a limited amount of data. Especially in areas where the dataset is limited (e.g. pox viruses), it has the capacity to produce very successful results.
- Low Configuration Requirement: End-to-end models require computers with high configuration. However, the model proposed in this research is capable of running on low-configuration machines.

In summary, our research provides a practical and efficient solution in terms of time, data and computational resources. In addition, the results show that our model is capable of competing with classical CNN models. The main advantages of our proposed method are given below:

- Monkeypox is an infectious disease, and to control the global spread of the virus, a new skin image dataset was collected to detect it via machine learning.

Fig. 10 Average classification accuracies according to the classifier methods

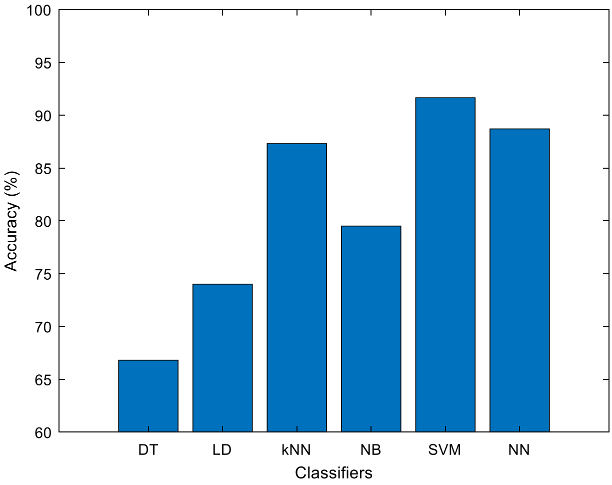

Table 8 Comparative results with the proposed model using a publicly available dataset

Method	Accuracy(%)
Ahsan et al. [45] for Case 1	83
Ahsan et al. [45] for Case 2	78
Our model for Case 1	96.67
Our model for four classes	94.74

- A transfer learning-based deep feature engineering model called MNPDenseNet is presented.
- To use the effectiveness of the patch-based models, new multiple patch-based algorithms. We employed a nested patch division. By using a nested patch division, the complexity of the patch-based feature generation since the nested-patch division generates less patch as compared with a fixed-size patch division.
- It is a self-organized computer vision model.
- The proposed MNPDenseNet attained a 91.87% classification accuracy with our new image dataset and a 94.74% accuracy with the publicly available dataset.
- Our model consistently outperformed the other computer vision methods.

The limitations of our proposed method are given below:

- There are limited images of skin afflictions by monkeypox on the web and in the literature. The publicly available monkeypox image dataset is small, with N = 171 including other classes. Although we collected N = 910 images from the web, these images are from five different categories. The number of images depicting monkeypox is small at N = 217.
- A larger dataset should be collected in the future to implement a real-time monkeypox detection model.

- A shallow classifier (SVM) was used to depict the classification capability of the generated features. However, an improved classifier can be used in this model, or the hyperparameters of the SVM can be optimized to obtain improved classification results.

5 Conclusions

A new image dataset related to monkeypox skin affliction was collected, and a new deep feature engineering architecture was proposed to detect monkeypox using these images. This model is named MNPDenseNet since multiple nested patches (six nested patch divisions) were used. Moreover, two feature extractors were added to this architecture, and these feature extractors were created using a pretrained DenseNet201. The three feature selectors employed were NCA, Chi2, and ReliefF. Our proposed MNPDenseNet yielded a 91.87% classification accuracy. When we applied MNPDenseNet to the publicly available dataset, our architecture yielded a 94.74% accuracy. Furthermore, our presented MNPDenseNet outperformed the other CNN models (AlexNet, MobileNetv2, DarkNet53, and ResNet50). In summary, the critical points about the proposed MNPDenseNet are as follows:

- The most appropriate initial patch size has dimensions 56 × 56.
- In our architecture, two feature generators were incorporated: the GAP layer and FC layer of the pretrained DenseNet201. The best feature extractor is GAP, among the feature extractors noted for solving the monkeypox classification problem.
- NCA, ReliefF, and Chi2 selectors were used in this research, and the best feature selection function was NCA.

These results and findings confirm the high classification success of the proposed MNPDenseNet model for monkeypox classification using skin images.

In the near future, we plan to acquire more skin images for monkeypox detection and apply our presented MNPDenseNet model to create a trained dataset. As a result, a new generation of automatic monkeypox detection desktop/mobile applications can readily be developed, and these applications can be introduced at medical centers to assist medical professionals. Moreover, attention-based deep networks can be used to obtain higher classification results.

Nomenclature **CNN**: Convolutional Neural Network; **FC**: Fully Connected; **GAP**: Global Average Pooling; **NCA**: Neighborhood Component Analysis; **Chi2**: Chi-Squared; **ReliefF**: Relief Feature Selection; **MNPDenseNet**: Multiple Nested Patch DenseNet; **IHMV**: Iterative Hard Majority Voting; **DT**: Decision Tree; **LD**: Linear Discriminant; **NB**: Naive Bayes; **kNN**: k-Nearest Neighbor; **SVM**: Support Vector Machine; **NN**: Neural Network; **OR**: Overall Recall; **OP**: Overall Precision; **OF1**: Overall F1-Score; **GM**: Geometric Mean; **CK**: Cohen's Kappa; **MCC**: Matthews Correlation Coefficient

Funding Open access funding provided by the Scientific and Technological Research Council of Türkiye (TÜBİTAK). The authors state that this work has not received any funding.

Data availability The dataset used in this study can be downloaded from [21, 46].

Declarations

Conflict of interest The authors of this manuscript declare no conflicts of interest.

References

1. Moss B (2013) Poxvirus DNA replication. Cold Spring Harbor Perspect Biology 5(9):a010199
2. Pal M, Mengstie F, Kandi V (2017) Epidemiology, Diagnosis, and Control of Monkeypox Disease: A comprehensive Review. Ame J Infect Diseases Microbio 5(2):94–9
3. Bethineedi LD, Kutikuppala LVS, Kandi V (2022) Monkeypox Epidemic: A Throwback From Smallpox Eradication. Cureus. 14(7)
4. Rao AK et al (2022) Use of JYNNEOS (Smallpox and Monkeypox Vaccine, Live, Nonreplicating) for Preexposure Vaccination of Persons at Risk for Occupational Exposure to Orthopoxviruses: Recommendations of the Advisory Committee on Immunization Practices—United States, 2022. Morbidity Mortality Weekly Rep 71(22):734
5. Bunge EM et al (2022) The changing epidemiology of human monkeypox—A potential threat? A systematic review. PLoS Neglected Tropic Dis 16(2):e0010141
6. Isidro J et al. (2022) Phylogenomic characterization and signs of microevolution in the 2022 multi-country outbreak of monkeypox virus. Nature Med, pp. 1-1
7. Hatch GJ et al (2013) Assessment of the protective effect of Imvamune and Acam 2000 vaccines against aerosolized monkeypox virus in cynomolgus macaques. J Virol 87(14):7805–7815
8. Magnus PV, Andersen EK, Petersen KB, Birch-Andersen A (1959) A pox-like disease in cynomolgus monkeys. Acta Pathologica Microbiologica Scandinavica 46(2):156–176
9. Breman JG, Steniowski M, Zanotto E, Gromyko A, Arita I (1980) Human monkeypox, 1970–79. Bullet World Health Organization 58(2):165
10. Abanobi O. Factors associated with the practice of monkey pox preventive behaviours among health workers in Yenagoa LGA, Bayelsa state, Nigeria
11. K. Chadaga et al. (2023) Application of Artificial Intelligence Techniques for Monkeypox: A Systematic Review. Diagnostics, 13. https://doi.org/10.3390/diagnostics13050824
12. Petersen E et al (2019) Monkeypox—Enhancing public health preparedness for an emerging lethal human zoonotic epidemic threat in the wake of the smallpox post-eradication era. Int J Infect Dis 78:78–84
13. Green JG, Durham TM, King TA (1988) 'University of Nebraska Medical Center, Omaha, Nebraska, USA;"Glenwood State Hospital and School. Dent 1:147–150
14. Zhang L, Gonzalez-Garcia A, Van De Weijer J, Danelljan M, Khan FS (2018) Synthetic data generation for end-to-end thermal infrared tracking. IEEE Trans Image Proc 28(4):1837–1850
15. Anitha A, Shivakumara P, Jain S, Agarwal V (2023) Convolution Neural Network and Auto-encoder Hybrid Scheme for Automatic Colorization of Grayscale Images. In: Smart Computer Vision: Springer, pp. 253-271
16. Dosovitskiy A et al. (2020) An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929
17. Tolstikhin I et al. (2021) MLP-Mixer: An all-MLP Architecture for Vision," arXiv preprint arXiv:2105.01601
18. Trockman A, Kolter JZ (2022) Patches are all you need?. arXiv preprint arXiv:2201.09792
19. Ahsan MM, Uddin MR, Farjana M, Sakib AN, Momin KA, Luna SA (2022) Image Data collection and implementation of deep learning-based model in detecting Monkeypox disease using modified VGG16. arXiv preprint arXiv:2206.01862

20. Sitaula C, Shahi TB (2022) Monkeypox virus detection using pre-trained deep learning-based approaches. J Med Syst 46(11):78
21. Ahsan MM, Uddin MR, Luna SA (2022) Monkeypox image data collection. arXiv preprint arXiv:2206.01774
22. Hussain MA, Islam T, Chowdhury FUH, Islam BMR (2022) Can artificial intelligence detect Monkeypox from digital skin images?. BioRxiv, 2022-08
23 Sahin VH, Oztel I, Yolcu Oztel G (2022) Human monkeypox classification from skin lesion images with deep pre-trained network using mobile application. J Med Syst 46(11):79
24. Ali SN et al. (2022) Monkeypox skin lesion detection using deep learning models: A feasibility study. arXiv preprint arXiv:2207.03342
25. Abdelhamid AA et al (2022) Classification of monkeypox images based on transfer learning and the Al-Biruni Earth Radius Optimization algorithm. Mathematics 10(19):3614
26. Bala D et al (2023) MonkeyNet: A robust deep convolutional neural network for monkeypox disease detection and classification. Neural Networks 161:757–775
27. UzunOzsahin D, Mustapha MT, Uzun B, Duwa B, Ozsahin, (2023) Computer-aided detection and classification of monkeypox and chickenpox lesion in human subjects using deep learning framework. Diagnostics 13(2):292
28. Yasmin F et al (2023) PoxNet22: A fine-tuned model for the classification of monkeypox disease using transfer learning. IEEE Access 11:24053–24076
29. Tuncer T, Ozyurt F, Dogan S, Subasi A (2021) A novel Covid-19 and pneumonia classification method based on F-transform. Chemo Int Labor Syst 210:104256
30. Kobat MA et al (2021) Automated COVID-19 and Heart Failure Detection Using DNA Pattern Technique with Cough Sounds. Diagnostics 11(11):1962
31. Barua PD et al (2021) Automatic COVID-19 detection using exemplar hybrid deep features with X-ray images. Int J Environ Res Pub Health 18(15):8052
32. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 4700-4708
33. Krizhevsky A, Sutskever I, Hinton GE (2017) Imagenet classification with deep convolutional neural networks. Commun ACM 60(6):84–90
34. Goldberger J, Hinton GE, Roweis S, Salakhutdinov RR (2004) Neighbourhood components analysis. Adv Neural Inform Proc Syst 17:513–520
35. Liu H, Setiono R (1995) Chi2: Feature selection and discretization of numeric attributes. In: Proceedings of 7th IEEE International Conference on Tools with Artificial Intelligence, IEEE, 388-391
36. Kira K, Rendell LA (1992) “A practical approach to feature selection,” in Machine learning proceedings. Elsevier 1992:249–256
37. Vapnik V (1998) The support vector method of function estimation. In: Nonlinear Modeling: Springer, 55-85
38. Vapnik V (2013) The nature of statistical learning theory. Springer science & business media
39. Kira K, Rendell LA (1992) The feature selection problem: Traditional methods and a new algorithm. Aaai 1992(2):129–134
40. Bagasta AR, Rustam Z, Pandelaki J, Nugroho WA (2019) Comparison of cubic SVM with Gaussian SVM: classification of infarction for detecting ischemic stroke. In: IOP Conference Series: Materials Science and Engineering, 546(5): IOP Publishing, p. 052016
41. Dogan A et al (2021) PrimePatNet87: prime pattern and tunable q-factor wavelet transform techniques for automated accurate EEG emotion recognition. Comput Biology Med 138:104867
42. Zhang Z, Schwartz S, Wagner L, Miller W (2000) A greedy algorithm for aligning DNA sequences. J Comput Bio 7(1–2):203–214
43. Powers DM (2020) Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. arXiv preprint arXiv:2010.16061
44. Warrens MJ (2008) On the equivalence of Cohen’s kappa and the Hubert-Arabie adjusted Rand index. J Class 25(2):177–183
45. Manjurul Ahsan M, M, Farjana M, Nazmus Sakib A, Al Momin K, Akter Luna S (2022) Image Data collection and implementation of deep learning-based model in detecting Monkeypox disease using modified VGG16. arXiv e-prints, p. arXiv: 2206.01862
46. M. Ahsan. "MONKEYPOX IMAGE DATA COLLECTION, https://github.com/mahsan2/Monkeypox-dataset-2022." (Accessed 01.09.2022).

Authors and Affiliations

Fahrettin Burak Demir[1] · Mehmet Baygin[2] · Ilknur Tuncer[3] · Prabal Datta Barua[4] · Sengul Dogan[5] · Turker Tuncer[5] · Chui Ping Ooi[6] · Edward J. Ciaccio[7] · U. Rajendra Acharya[8,9]

✉ Sengul Dogan
sdogan@firat.edu.tr

Fahrettin Burak Demir
fdemir@bandirma.edu.tr

Mehmet Baygin
mehmet.baygin@erzurum.edu.tr

Ilknur Tuncer
ilknur.tuncer@icisleri.gov.tr

Prabal Datta Barua
prabal.barua@usq.edu.au

Turker Tuncer
turkertuncer@firat.edu.tr

Chui Ping Ooi
cpooi@suss.edu.sg

Edward J. Ciaccio
ciaccio@columbia.edu

U. Rajendra Acharya
Rajendra.Acharya@usq.edu.au

[1] Department of Software Engineering, Faculty of Engineering and Natural Sciences, Bandirma Onyedi Eylul University, Bandirma, Turkey

[2] Department of Computer Engineering, College of Engineering, Erzurum Technical University, Erzurum, Turkey

[3] Elazig Governorship, Interior Ministry, Elazig, Turkey

[4] School of Business (Information System), University of Southern Queensland, Queensland, Australia

[5] Department of Digital Forensics Engineering, Technology Faculty, Firat University, Elazig, Turkey

[6] School of Science and Technology, Singapore University of Social Sciences, Singapore 599494, Singapore

[7] Department of Medicine, Columbia University Irving Medical Center, Columbia, USA

[8] Centre for Health Research, University of Southern Queensland, Queensland, Australia

[9] School of Mathematics, Physics and Computing, University of Southern Queensland, Springfield, Australia