Niğde Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi
Niğde Ömer Halisdemir University Journal of Engineering Sciences

Araştırma makalesi / Research article

www.dergipark.org.tr/tr/pub/ngumuh / www.dergipark.org.tr/en/pub/ngumuh

Derin öğrenme yöntemleri kullanılarak deri lezyon görüntülerinden Maymun Çiçeği hastalığının tespiti

Detection of Monkeypox disease from skin lesion images using deep learning methods

Muhammet Talha Engin [1], **Kemal Adem** [2,*]

[1] *Aksaray Üniversitesi, Sosyal Bilimler Enstitüsü, Yönetim Bilişim Sistemleri Anabilim Dalı, 68100, Aksaray Türkiye*
[2] *Sivas Cumhuriyet Üniversitesi, Bilgisayar Mühendisliği, 58100, Sivas, Türkiye*

Abstract

Monkeypox is a disease that, while less deadly and contagious than COVID-19, could pose a global pandemic threat. In the field of medical imaging, deep learning techniques offer promising results in the diagnosis of diseases. This study develops deep learning models using skin lesion images for early diagnosis of monkeypox. The research is divided into two key sections. In the first section, a deep learning model is developed using the Monkeypox Skin Image Dataset (MSID). The second section focuses on a model trained on a combined dataset, which merges the Monkeypox Skin Image, Monkeypox Master, and Monkeypox Original Image Datasets, referred to as HYBRID. The MSID dataset comprises 806 Monkeypox and 690 Non-Monkeypox images for training, along with 309 Monkeypox and 292 Non-Monkeypox images for testing, resulting in a total of 2,097 images of skin lesions with and without monkeypox. The HYBRID dataset includes 1,088 Monkeypox and 1,896 Non-Monkeypox images for training, as well as 468 Monkeypox and 812 Non-Monkeypox images for testing, resulting in a total of 4,264 skin lesion images. Five distinct deep learning models—DenseNet201, InceptionResNetV2, InceptionV3, NASNetLarge, and Xception—were applied to both datasets, and the outcomes were compared. The DenseNet201 model, when trained on augmented data, demonstrated remarkable performance in detecting monkeypox, achieving accuracy rates of 99.33% on the MSID dataset and 98.52% on the HYBRID dataset.

Keywords: Monkeypox Disease, Virus, Deep Learning, DenseNet201

Öz

Monkeypox, COVID-19 salgınına kıyasla daha az ölümcül ve bulaşıcı olsa da küresel bir salgın tehdidi oluşturabilen bir hastalıktır. Tıbbi görüntüleme alanında derin öğrenme teknikleri, hastalıkların teşhisinde umut verici sonuçlar sunmaktadır. Bu çalışma, monkeypox'un erken teşhisi için deri lezyon görüntülerini kullanarak derin öğrenme modelleri geliştirir. Çalışma iki ana bölümden oluşmaktadır. İlk bölüm, Monkeypox Deri Görüntü Veri Seti (MSID) kullanılarak eğitilen bir derin öğrenme modelini tanıtırken, ikinci bölüm Monkeypox Deri Görüntü, Monkeypox Master ve Monkeypox Orjinal Görüntü Veri Setlerinden (HYBRID) oluşturulan birleştirilmiş bir veri seti ile eğitilen bir modeli sunar. MSID veri seti, toplamda 2097 resim içermektedir. Bu set, eğitim için 806 Monkeypox ve 690 Non Monkeypox, test için ise 309 Monkeypox ve 292 Non Monkeypox deri lezyon görüntüsünden oluşmaktadır. HYBRID veri kümesi ise toplamda 4264 resim içermektedir. Bu set, eğitim için 1088 Monkeypox ve 1896 Non Monkeypox, test için ise 468 Monkeypox ve 812 Non Monkeypox deri lezyon görüntüsünü barındırmaktadır. Bu veri setlerinin eğitiminde beş farklı derin öğrenme modeli- DenseNet201, InceptionResNetV2, InceptionV3, NASNetLarge ve Xception- bu iki veri seti üzerinde kullanılarak sonuçlar karşılaştırılmıştır. Artırılmış veri setleri ile eğitilen DenseNet201 modeli, monkeypox'u tanıma konusunda önemli başarılar elde etti ve sırasıyla MSID ve HYBRID veri setlerinde %99.33 ve %98.52 doğruluk oranlarına ulaştı.

Anahtar kelimeler: Maymun çiçeği hastalığı, Virüs, Derin öğrenme, DenseNet201

1 Introduction

Epidemics are diseases that spread independently of time and space with a high infection rate [1]. Historically, it is known that diseases such as cholera, plague, tuberculosis and leprosy spread throughout the world and to various regions [2]. The geographical location, natural beauties and riches of our country create a favorable environment for the emergence and spread of epidemic diseases. Epidemics caused by different reasons such as wars, commercial activities and migration have led to the death of millions of people [3]. In recent years, we are facing new epidemics worldwide, such as COVID-19 and Monkeypox. COVID-19 is a disease that affects the respiratory tract and has become more dangerous than other epidemics [4]. Monkeypox is a viral zoonotic disease that causes a rash similar to smallpox, but with a lower spread rate and mortality [5]. Both diseases cause serious health problems. Monkeypox is a disease caused by the monkeypox virus and is related to smallpox.

* Sorumlu yazar / Corresponding author, e-posta / e-mail: kemaladem@cumhuriyet.edu.tr (K. Adem)
Geliş / Recieved: 14.02.2024 Kabul / Accepted: 12.08.2024 Yayımlanma / Published: 15.10.2024
doi: 10.28948/ngumuh.1436907

The virus is usually transmitted through contact with infected animals, but humans and monkeys can also be incidental hosts. Monkeypox virus can be transmitted through contact with skin and mucous membrane lesion fluid, animal saliva, biting/scratching, contact with urine or feces, or animal bites. The potential for human-to-human transmission is low, but cases have been reported in non-endemic countries due to factors such as direct contact with infected lesions during sexual contact. Monkeypox can also be transmitted by droplet transmission, especially during face-to-face contact at close quarters without personal protective equipment. The risk of transmission increases with increasing exposure at a distance of less than 2 meters for at least 3 hours. Monkeypox virus has been assessed as a disease with epidemic potential. Measures have been taken around the world to contain the outbreak. The World Health Organization declared many African countries as endemic areas and declared a global public health emergency [6]. Technology played an important role in improving transportation and communication channels during the pandemic. Technological developments are of great importance in planning health services and controlling the epidemic. Advances in medicine and technology are reducing the impact of epidemics and controlling their spread. Technological advances such as deep learning methods have been used in areas such as image classification and object detection. Conventional methods for detecting monkeypox require contact. This study uses machine learning and deep learning methods and will contribute to the literature by detecting the disease without requiring physical contact. Detection of the disease without physical contact will reduce the spread rate. It is hoped that these advances will continue in the future to develop more effective outbreak prevention and treatment strategies. [7-12].

2 Literature review

Nowadays, the field of deep learning has gained great importance and various studies have been carried out in this field. The analysis and evaluation of these studies will allow us to understand the current state of knowledge and benefit from the results described in this field. In this section, we present an overview of previous research on the detection of monkeypox using deep learning methods. While reviewing the related literature, we aimed to find results obtained in similar studies using image processing to guide our work.

Chiranjibi and Tej obtained Precision: 85.44%; Recall: 85.47%; F1-score: 85.40%; and Accuracy: 87.13% in their study on monkeypox images using 13 different pre-trained deep learning models: 85.40%; and Accuracy: 87.13%. He also stated that his study was conducted on limited data and that the performance will be further improved by increasing the dataset.

Shams et al. used ResNet50, VGG16 and InceptionV3 models in their study on a dataset consisting of 1428 and 1764 images after enlargement with 3 different trained models in the "Monkeypox Skin Lesion Dataset (MSLD)" consisting of 102 monkeypox skin lesions and 126 other datasets, and the highest reliability rate was obtained with ResNet50 (82.96±4.57%). With the mobile application developed with deep learning support, it has been revealed that it may have the potential to detect monkeypox disease.

Sahin et al. trained the data classified as positive and negative using TensorFlow. The mobile application now incorporates the TensorFlow Lite model alongside its library for detecting monkeypox. This integrated system allows individuals with bodily lesions to promptly conduct an initial diagnosis. Consequently, those potentially afflicted with monkeypox are prompted to seek specialist attention promptly for confirmation. Test outcomes indicate that the system achieved a classification accuracy of 91.11% for images.

Md et al. Since there is no publicly available Monkeypox dataset for ML model development, the study presents a new dataset named "Monkeypox2022" with images obtained from many open source and online platforms without any restrictions, including commercial purposes. The study proposes a modified VGG16 model that includes two separate studies, the first study and the second study. Exploratory calculations show that the proposed model can detect Monkeypox patients with high accuracy rates. The accuracy rate for Study 1 was 97 ± 1.8% (AUC = 97.2) and for Study 2 was 88 ± 0.8% (AUC = 0.867). Furthermore, when explaining the model's predictions and feature extraction, he used Local Explainable Model-Independent Explanations (LIME) to provide deeper insight into the specific features that characterize the onset of Monkeypox virus. In this study, he demonstrated the potential of an ML-based approach for the diagnosis of Monkeypox disease and emphasized the need for further research and the creation of datasets.

Md et al. aimed to integrate deep transfer learning approaches with a convolutional block attention module (CBAM) for the image-based classification of human monkeypox. They utilized five deep learning models, namely VGG19, Xception, DenseNet121, EfficientNetB3, and MobileNetV2, incorporating attention mechanisms focused on both channels and spatial dimensions. A comparative analysis was performed across these models. The Xception-CBAM-Dense architecture stood out, surpassing the others in accurately classifying human monkeypox and related conditions, achieving a validation accuracy of 83.89%.

Altun et al. evaluated the performance of several deep learning models—including MobileNetV3-s, EfficientNetV2, ResNet50, VGG19, DenseNet121, and Xception—using metrics such as AUC, accuracy, sensitivity, loss, and F1-score for comparison. The optimized hybrid MobileNetV3-s model stood out by delivering the best results, with an average F1-score of 0.98, an AUC of 0.99, an accuracy of 0.96, and a sensitivity of 0.97. Their research involved using convolutional neural networks along with hyperparameter optimization and a custom hybrid function transfer learning approach to develop a specialized CNN model, which achieved impressive performance.

In his research titled "A vision transformer-based approach for automatic COVID-19 diagnosis from chest X-ray images," Pacal introduced a diagnostic framework leveraging the transformer method, which represents the latest and notably more popular deep learning architecture compared to previous techniques like CNN-based approaches. This system incorporates vision transformer models to enhance the diagnosis of COVID-19 on a novel dataset, COVID-QU-Ex. Through experimental analysis, Pacal noted the superior performance of vision transformer models over CNN models. Specifically, the ViT-L16 model exhibited a test accuracy and F1-score exceeding 96%, showcasing significantly enhanced performance compared to similar studies documented in existing literature.

Akyel, in his study titled Diagnosis of oral cancer from histopathologic images with Xception, used the Xception model pre-trained and modified with the sequential algorithm unlike the existing literature. In the training phase with two classes of normal and oral squamous cell cancers, he achieved 98.70% training success, 97.20% membrane accuracy, 96.50% precision and 97% sensitivity respectively. These values were compared with some other studies in the literature. In his study, it was seen that the Xception model together with Clahe is a suitable option in oral cancer classification and can be useful in the diagnosis process.

Diker presents the performance comparison of deep learning techniques in cerebral palsy classification in his study titled Automatic classification of cerebral palsy using deep learning methods. The performance of different deep learning algorithms such as EfficientNetB0, MobileNet-V2, ResNet18 and VGG-16 are compared respectively. EfficientNetB0, MobileNet-V2, ResNet18 and VGG-16 obtained 84.6%, 87.4%, 80.9% and 92.6% accuracy values for deep learning models respectively.

Çeçen and Özer, in their study titled Classification of pathological breast cancer images with YOLOv5 algorithm, pathological images of biopsies taken from breast tissue were analyzed with deep learning method, a sub-branch of artificial intelligence. Classification of breast cancer type was studied. YOLOv5 algorithm was used as a deep learning model. As a result of the experiments, tumor classification was achieved with a high success rate of 95.3% in the accuracy parameter.

In Tüfekçi and Gezici's study titled "Detection of COVID-19 and viral pneumonia from chest x-ray images with deep learning," the aim was to establish a highly accurate and dependable model capable of discerning between COVID-19 and viral pneumonia based on chest x-ray images. The research involved a comprehensive modeling endeavor, employing both the original and transfer learning-enhanced versions of AlexNet and GoogleNet, specialized architectures of Convolutional Neural Networks, renowned in deep learning. The dataset utilized in the modeling process, the COVID-19 Radiography Database, is a widely used yet imbalanced dataset featuring three classes with varying sample sizes. To address this, data reduction and augmentation techniques were employed, resulting in the creation of two balanced datasets with an equal number of samples in each class. The original dataset and the newly balanced datasets were partitioned into training and test sets in an 80-20 ratio and further divided into subsets for 3, 5, and 10-fold cross-validation. Model performances were evaluated, and the model with the highest performance was identified. Ultimately, the most effective model was determined to be the one employing the AlexNet architecture with transfer learning on the augmented dataset, segmented into 10-fold cross-validation, achieving an accuracy of 99.90%.

According to Inik and Ulker, the extraction of feature vectors is of great importance in building machine learning systems. However, this process is time-consuming as it usually requires the intervention of experts in the field and they cannot process the raw data directly. Unlike classical machine learning techniques, deep learning solves this problem by learning on raw data. Deep networks obtain the necessary information by creating representations in different layers.

Deep learning has attracted attention with its success in object classification, especially in ImageNet, a large-scale visual recognition competition organized in 2012. The reason behind this success lies in the availability of sufficient data for training and the hardware infrastructure to process this data. Convolutional Neural Network (DNN) architecture includes convolution, pooling, ReLU, DropOut, fully connected and classification layers. These architectures constitute the basic building blocks of deep learning. In addition, deep learning architectures such as AlexNet, ZFNet, GoogLeNet, Microsoft ResNet and RCNN have also played an important role in the development of deep learning.

Kumar et al. discussed the limitations of RT-PCR tests commonly used for Covid-19 diagnosis, highlighting their propensity for false positive and false negative results and noting the lengthy process involved. To improve accuracy, they suggested increasing the number of simultaneous tests. Additionally, they pointed out that CT scans and X-rays could detect Covid-19-related pneumonia early on. By applying advanced deep learning techniques, they achieved over 95% accuracy in diagnosing Covid-19. The team employed eight deep learning models based on convolutional neural networks (CNN), including ResNet 152 v2, InceptionResNet v2, Xception, Inception v3, ResNet 50, NASNetLarge, DenseNet 201, and VGG 16, to detect pneumonia in both X-ray and CT scan images. Comparative analysis indicated that these models effectively differentiate Covid-19 positive cases.

On a related note, Ahmet et al. described oral cancer as a serious disease, ranking as the seventh most common cancer globally. Histopathologic examination is the current gold standard for diagnosing oral cancer, especially in the head and neck regions. However, this method is time-consuming and demands expert analysis. Timely identification of Oral Squamous Cell Carcinoma (OSCC) is vital for successful treatment, reducing mortality and morbidity rates, and enhancing patient survival prospects. To aid clinicians, artificial intelligence techniques are being explored to reduce the workload on pathologists significantly. Ahmad et al. developed hybrid methods combining different features to enhance early OSCC diagnosis. Their research employed three different strategies, each using five unique models. The first strategy implemented transfer learning with models such as Xception, Inceptionv3, InceptionResNetV2, NASNetLarge, and DenseNet201. The second strategy combined feature extraction from pre-trained CNN models with Support Vector Machines (SVM) for classification. Features extracted from models like Xception, Inceptionv3, InceptionResNetV2, NASNetLarge, and DenseNet201 were processed through the SVM algorithm for classification accuracy evaluation. The third strategy employed a hybrid feature fusion technique, combining deep features from the previously mentioned models. These features were reduced in dimensionality using principal component analysis (PCA) and merged with shape, color, and texture features obtained through the gray level co-occurrence matrix (GLCM), Histogram of Oriented Gradient (HOG), and Local Binary Pattern (LBP) methods. This hybrid feature fusion was integrated into an SVM model to improve classification accuracy. The proposed system showed impressive results for the rapid diagnosis of OSCC using histological images, with the SVM based on hybrid feature fusion achieving 97.00% accuracy, 96.77% precision, 90.90% sensitivity, 98.92% specificity, a 93.74% F-1 score, and an AUC of 96.80%.

3 Material and method

The database of our research consists of two parts. In the first part, the Monkeypox Skin Image Dataset (MSID) dataset was used on the Kaggle site. In the second part, the HYBRID dataset, which consists of Monkeypox Skin Image Dataset (MSID) on Kaggle, Monkeypox Master (MM) on GitHub and Monkeypox Original Images (MOI) on Kaggle, was used. After all images were collated, similar images were identified and extracted with the Visipics application. The datasets were preprocessed by inverting the colors and converting them to brown and gray scale, and data augmentation was performed in the same way in both datasets. In addition, random rotation (rotation_range), random zooming (zoom_range), random horizontal shift (width_shift_range), random vertical shift (height_shift_range), random vertical shift (height_shift_range), random horizontal shift (shear_range), random horizontal symmetry (horizontal_flip) and fill_mode (fill_mode) preprocessing were applied to all datasets.

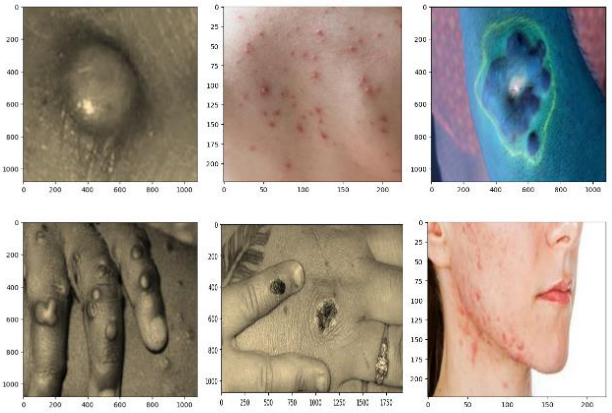


Figure 1. Examples of datasets with Monkeypox-MSID

As a result, the MSID cluster consists of 281 original images diagnosed with monkeypox, 278 color inverted images, 278 brown-scale images and 278 gray-scale images. The non-monkeypox images in the MSID set consist of 247 original images, 245 color inverted images, 245 brown scale images and 245 gray scale images. Data samples for this dataset are shown in Figure 1.

The HYBRID dataset consists of 389 original images with monkeypox diagnosis, 389 color inverted, 389 brown-scaled and 389 gray-scaled images. The non-monkeypox images in the HYBRID cluster consist of 677 original images, 677 color inverted, 677 brown-scaled and 677 gray-scaled images. Data samples of this dataset are shown in Figure 2.

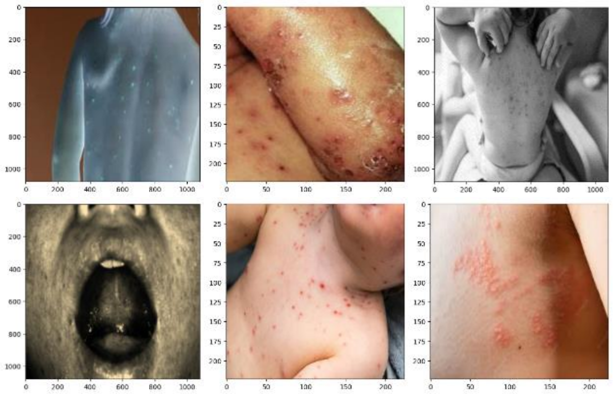


Figure 2. Examples of datasets with Monkeypox-HYBRID

While these datasets were divided into training and test files, 70% of training and 30% of test images were randomly generated with the code generated in Python. The data set distribution is shown in Table 1.

Table 1. Number of ımages in datasets

		Train	Test
Image of Number	MSID	1496	601
	HYBRID	2984	1280

There are many models commonly used in deep learning methods. In this study, DenseNet201, InceptionResNetV2, InceptionV3, NASNetLarge and Xception models, which are widely used in Convolutional Neural Networks, are used. The important advantage of these models is that they have improved information flow, parameter efficiency and the highest accuracy rates. In this respect, DenseNet201, InceptionResNetV2, InceptionV3, NASNetLarge and Xception models were preferred in this study.

3.1 Convolutional neural networks (CNN)

In 1998, Bottou, LeCun, Haffner, and Bengio introduced the LeNet model, one of the pioneering examples of convolutional neural networks. The LeNet model, which was initially used only to identify postal codes and numbers, has now become a model that calculates billions of parameters, extracts meaning from data and finds solutions to various problems [13]. Convolutional neural networks have made significant progress in computer vision, image-based applications and other fields. It has been observed that it has brought great success to many different fields such as face recognition, object identification, image categorization, autonomous construction, and disease detection [13, 14]. Convolutional neural networks (CNNs or ConvNets) are deep learning models designed to process input images and identify various objects within them [15]. CNNs share similarities with the architecture of artificial neural networks, featuring a multilayer structure. They comprise a convolutional layer, a pooling layer, and a fully connected layer. The convolutional and pooling layers are responsible for extracting features from the input image, while the fully connected layer forwards these features to the output layer for classification [16]. Figure 3 visually shows the structure of these layers. The fully connected layer prepares the classification process by connecting the neurons in the first layer to the neurons in the next layer [12]. [18].

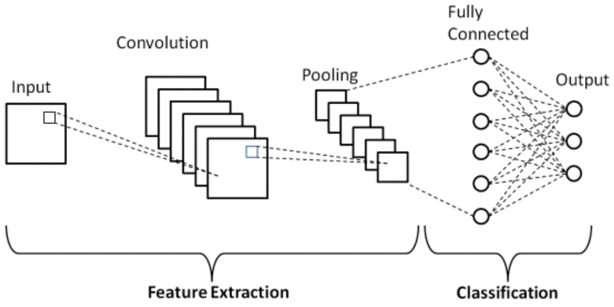
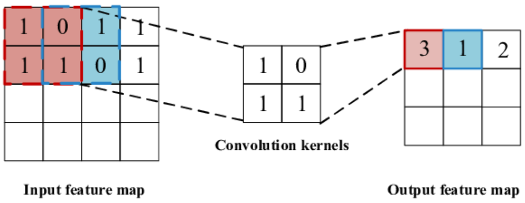


Figure 3. Basic structure of a Convolutional Neural Network

3.1.1 Convolution layer

It includes components called filters and feature maps, which determine the basic architecture of CNN and where a large computational process takes place [13]. Filters in this layer are used to detect features on the input image and play an important role in the creation of feature maps [19]. In this layer, the transformation process is performed using filters that are smaller in size than the input image. Filters are usually 2x2, 3x3 or 5x5 in size and perform a shift operation on the entire image [15]. These operations result in feature maps, which summarize the important information in the input image [16]. Figure 4 shows the output image obtained by applying a 3x3 convolution filter to a 5x5 matrix image given as input. This filter is applied by shifting over the image and this process is performed over the entire image [20].



1	0	1	1
1	1	0	1

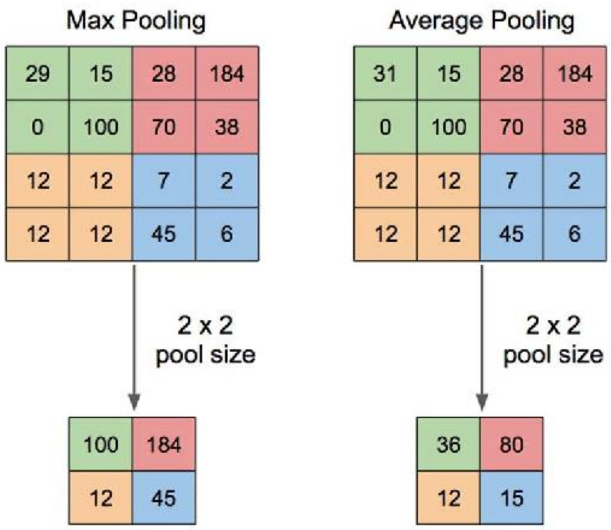
1	0
1	1

3	1	2

Figure 4. Image Convolution and output generation Process

3.1.2 Pooling Layer

The pooling layer is a layer that may not be used in some models because its use is not mandatory. In the pooling layer, computations are performed quickly and feature attributes from the previous layer are grouped to create new feature maps [19]. This layer aims to reduce the size for the next layer without loss of features. However, the reduction in the process may result in information loss. Although information loss may seem disadvantageous, it mianly helps to avoid overcompliance learning by reducing the computational load for the next layers [15]. Therefore, information loss is a feature which can be turned into an advantage of the system.



Max Pooling

29	15	28	184
0	100	70	38
12	12	7	2
12	12	45	6

2 x 2 pool size

100	184
12	45

Average Pooling

31	15	28	184
0	100	70	38
12	12	7	2
12	12	45	6

2 x 2 pool size

36	80
12	15

Figure 5. Maximum and average values of Pooling Layers [35]

The pooling layer is divided into two different structures, the maximum pooling layer and the average pooling layer, which are shown in Figure 5. Maximum pooling is a widely preferred method as it generally performs better. In this layer, the highest value of the pixels in the region scanned by the filters is selected as the output [13]. On the other hand, in the average pooling method, the average of the values in the selected region during the scanning process is determined as the output and the same process is applied to the other cells in the new array [19].

3.1.3 Activation functions

The choice of activation function depends on the neural network model used and the problem to be solved. Each activation function has different advantages and properties. Here are the preferred activation functions for some situations:

The sigmoid function is commonly used in the output layer of neural networks. It is especially preferred for binary classification problems. The sigmoid function allows us to interpret the output as a probability value between 0 and 1.

The Tanh function is similar to the sigmoid function but limits the output between -1 and 1. Common uses are binary classification problems where outputs with a range of -1 to 1 are needed, or when symmetric activation functions are required.

ReLU (Rectified Linear Unit) is an activation function commonly used in deep learning models. It has advantages such as fast computation and the ability of the model to learn faster. However, the fact that the ReLU function takes negative input values as zero can cause over-activation problems in some cases.

Leaky ReLU is an extension of the ReLU function. It preserves negative values by leaking on negative input values. This can help alleviate the over-activation problem.

The softmax function is used in multi-class classification problems. It allows us to compare the probabilities between classes by converting values from multiple outputs into a probability distribution.

Choosing an activation function depends on the dataset characteristics, the problem type, and the model's performance. You can choose the most appropriate activation function by conducting experiments and optimizing your model [21].

Deep learning models use components such as layers, optimization algorithms and loss functions along with activation functions to generate output based on input data. ReLU is a nonlinear activation function that is widely used in the literature [22]. ReLU is a function that performs well in problems where linearity is not required in the learning process [23]. The ReLU function is mathematically defined in Equation 1 [17].

$$f(x) = \max(0, x) \tag{1}$$

Here *x* represents the input value and the output of the function takes the value *x* if it is greater than 0 and 0 if it is less than 0.

ReLU (Rectified Linear Unit) is the preferred activation function in neural networks and has several advantages:

Fast Computation: ReLU is computationally faster compared to other activation functions such as sigmoid or tanh. This provides a significant advantage in computationally intensive applications such as large data sets or deep neural networks.

Linearity: ReLU exhibits a linear relationship for positive input values. Being a linear function gives the model better learning ability and makes it easier to update the weights.

No Decay Problem: The ReLU function produces a linear output for positive input values and zero output for negative input values. This ensures the correct transmission of the gradient during the training of the network and avoids the vanishing gradient problem.

Sparse Activation: ReLU assigns negative input values as zero. Therefore, a neural cell usually triggers only a few activations. This makes the network work in a more organized and less dependent way and reduces the risk of overfitting.

However, the fact that it outputs completely zero for negative input values can lead to a dead neuron problem in some cases [24]. In this case, modifications such as Leaky ReLU can be used. Since ReLU is a widely preferred activation function in deep learning models due to these advantages, we preferred this function in our study.

3.1.4 Optimization method

Optimization is the process of optimally minimizing or maximizing a cost or loss function under certain constraints and parameters. In deep learning applications, it is known that there is an optimization problem underlying the learning process. In solving this problem, various optimization methods are used to discover nonlinear relationships and determine the optimal value. Various optimization algorithms can affect performance and speed in deep learning models. Among these algorithms, algorithms such as stochastic gradient descent, adadelta, adagrad, adamax, adam are frequently preferred [25]. These algorithms aim to minimize the loss function of the model using different optimization strategies and update rules. In this study, the Adam optimization algorithm is preferred due to its recent popularity and effectiveness in deep learning applications. Adam performs gradient-based optimization using adaptive moment estimation. This algorithm is known for its ability to quickly approach the minimum of the loss function and its adaptive properties in parameter updates. Using the Adam optimization algorithm is an effective choice for enhancing model performance and accelerating the learning process.

3.1.5 Fully connected layer

In deep learning models, particularly convolutional neural networks (CNNs), it is common practice to convert multidimensional data into one-dimensional sequences. This process is called "flattening" and is usually performed before the classification layer. Figure 6 shows an example of the flattening process. On the left side of Figure 6, a "Pooled Feature Map" is shown. This is a 3x3 matrix obtained after pooling in a CNN layer. The elements of this matrix are shown as 1, 1, 0 in the first row, 4, 2, 1 in the second row and 0, 2, 1 in the third row, respectively. The smoothing operation aims to transform the elements of this multidimensional matrix into a one-dimensional array. As shown in the figure, the elements of the matrix are taken in order (left to right and top to bottom) to form a one-dimensional array of 1, 1, 0, 4, 2, 2, 1, 0, 2, 1. This process is usually performed from row to row.

The smoothing process allows the multidimensional data generated by the intermediate layers of the CNN to be transferred to the fully connected layers. The fully connected layers perform the final classification process using this one-dimensional sequence. This process is critical in bringing the output of the network into a format suitable for classification. Thanks to the smoothing process, CNNs with multidimensional data structures can perform classification by transferring the obtained feature maps to fully connected layers. This process is a crucial step for deep learning models to function effectively, as illustrated in Figure 6, makes the input data suitable for the networks used.

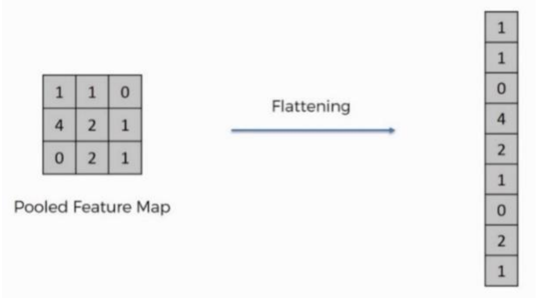


Figure 6. Data flattening in the fully connected layer for data transfer

3.2 *DenseNet201 architecture*

DenseNet201 is a deep learning model characterized by its neural network architecture with 201 layers of densely connected blocks. It is available as a pre-trained model for various tasks, including image classification, object detection, and segmentation. DenseNet201 is built on a convolutional neural network (CNN) architecture and is characterized by densely connected blocks. Densely connected blocks are designed to increase the depth of the network and provide better sharing of features. This architecture was developed by Huang et al. to enable knowledge transfer. For each layer, the outputs of the previous layers are used as input values. DenseNet is a network architecture that helps to significantly reduce the vanishing gradient problem. To achieve this, DenseNet has a reduced number of parameters [26]. DenseNet201 has a total of 201 layers consisting of an initial layer starting with a 7x7 convolution filter followed by a 3x3 MaxPool, 196 convolution layers with 3 average pool layers in between, and 1 output layer [27]. The DenseNet201 architecture is given an input of size 224×224×3. In this model, each densely connected block is formed by combining the outputs of all layers in the previous block [28]. These dense connections facilitate the flow of information and allow the network to learn better features at deeper layers.

DenseNet201, trained on the ImageNet dataset, has demonstrated high performance in general image classification tasks. It is also adaptable for various applications through transfer learning methods. As a robust model in deep learning, DenseNet201 is frequently employed for complex and multi-class datasets. The architecture of the DenseNet201 model is illustrated in Figure 7.

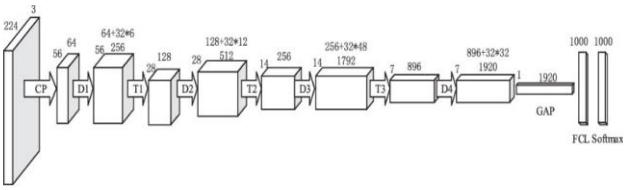


Figure 7. DenseNet201 architecture

3.3 *NasNetLarge architecture*

The NasNetLarge model was introduced by Zoph in the paper "Learning Transferable Architectures for Scalable Image Recognition". This work includes research on the automatic design of a customized convolutional neural network (CNN) architecture using an artificial intelligence algorithm called Neural Architecture Search (NAS). The NasNetLarge model in Figure 8 is based on a network structure generated by the NASNet algorithm and optimized for large-scale image classification tasks. The model was developed by Google using the TensorFlow library. Furthermore, the paper describes in detail the dataset used during the training of the model and the training process. The NasNetLarge model is often used for image classification, object recognition and similar tasks using Transfer Learning. It can be fine-tuned for different datasets and tasks using pre-trained weights [29].

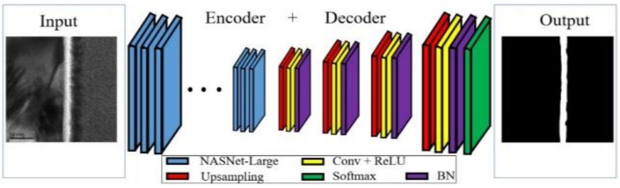


Figure 8. NasNetLarge architecture

3.4 *InceptionV3 architecture*

The Inception model is a CNN architecture that achieved high classification accuracy on the ImageNet dataset and won first place in the ILSVRC 2014 competition. This model has achieved notable success in image classification, significantly reducing the error rate compared to other competitors. In particular, it has a lower error rate than the VGG deep learning model introduced in 2014 [30]. The achievements with the Inception model have highlighted areas where accuracy can be improved and where model complexity needs to be reduced. In realizing these improvements, different versions of the Inception model have been developed and the InceptionV3 model is one of these versions. The InceptionV3 model builds on the original Inception model with the goal of further enhancing accuracy [31].

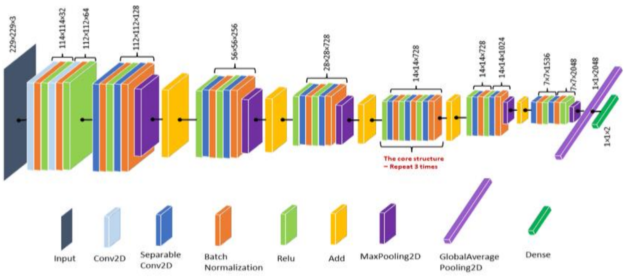


Figure 9. InceptionV3 architecture

The InceptionV3 model is a model that has a depth of 48 layers and uses inception modules that combine convolutions of dimensions 1x1, 3x3, 5x5. This model is known to be more efficient than VGGNet. The architecture of the InceptionV3 model is depicted in Figure 9. InceptionV3 uses two/three 3x3 filters for convolutions of size 5x5 or 7x7. This has been observed to increase the speed of the training process as the number of parameters is reduced [30]. It is known that as the depth of the model increases, its width also increases. Therefore, in the InceptionV3 model, it is aimed to provide high information flow by optimizing the width and depth of the network [32]. As a result, the InceptionV3 model is a model with a depth of 48 layers and uses inception modules that combine convolutions of different sizes. In this model, the number of parameters is reduced by optimizing the size and number of filters in the convolutions and high information flow is achieved. In this way, the InceptionV3 model has a more efficient structure than VGGNet.

3.5 InceptionResNetV2 architecture

InceptionResNetV2 is a deep learning model developed by Google and was detailed in the 2016 article "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning" by Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi [33]. This model combines the Inception architecture with residual connections. The Inception architecture aims to optimize the flow of information by using "inception" modules that combine convolutional operations at different scales. In this way, the model is more efficient to detect features at different scales. Furthermore, residual connections allow the model to learn deeper and more effectively. The InceptionResNetV2 model has a depth of 48 layers and uses inception modules with convolution operations in dimensions 1x1, 3x3, 5x5. These modules make it possible to detect various features by combining different filter sizes. In addition, having fewer parameters and preferring 3x3 filters increases the training speed and allows the model to learn more effectively. The InceptionResNetV2 model has a depth of 48 layers and uses inception modules with convolution operations in dimensions 1x1, 3x3, 5x5. These modules make it possible to detect various features by combining different filter sizes. In addition, having fewer parameters and preferring 3x3 filters increases the training speed and allows the model to learn more effectively. The InceptionResNetV2 model is frequently utilized in visual processing tasks, including image classification, object detection, and face recognition. Especially when trained on large datasets such as ImageNet, this model achieves high success in detecting and classifying various image features. It can also be used in different visual processing problems with transfer learning methods to achieve impressive results. The InceptionResNetV2 model represents a significant advancement in deep learning and has demonstrated successful results across various application areas. The InceptionResNetV2 architecture model is shown in Figure 10.

3.6 Xception architecture

The Xception model is a deep learning architecture introduced by François Chollet in his 2017 paper, "Xception: Deep Learning with Depthwise Separable Convolutions.". Xception is a convolution-based neural network and is based on the Inception architecture. In the paper, depthwise separable convolutions, which is a different structure from the traditional convolution process, is used to increase the learning capacity and computational efficiency of the model. Unlike traditional convolutions, depthwise separable convolutions use a two-stage approach. In the first stage, convolution is performed on the input channels and then channel-level convolutions are processed separately. This structure reduces the number of parameters while preserving the network's learning ability and reducing the computational cost. The Xception model has achieved particularly effective results in image classification and visual processing tasks. Through the use of depth-structured convolutions, better learning performance is achieved while requiring fewer computational resources. The xception model is available in the Keras library as a module called "xception". Keras provides a tool that facilitates the use of this model and supports customization. The Xception model can be used in different image processing problems with transfer learning methods using pre-trained weights [34].

Xception has been a significant advance in the field of deep learning and has become an effective model thanks to the use of depth-structured convolutions. High classification accuracy and computational efficiency have made Xception popular in different application areas.

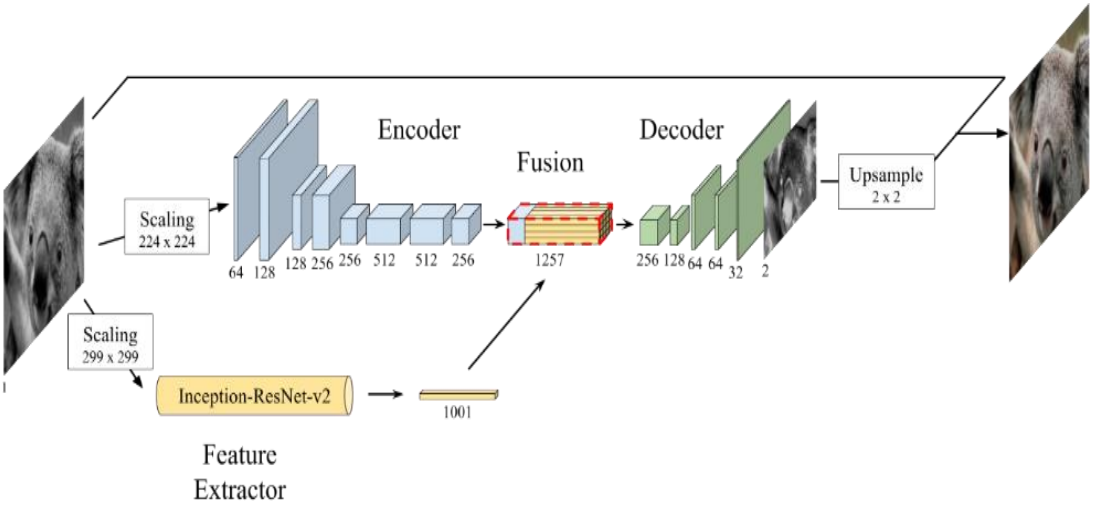


Figure 10. InceptionResNetV2 architecture

4 Findings and discussion

The proposed methods utilized the Google Colab Pro environment and GPU/TPU support and the Jupyter notebook due to the lack of an existing powerful machine.

Performance metrics were needed to test the success of the methods for monkeypox detection. Performance measures are metrics used to evaluate and compare the success rates of different proposed methods. In order to know these metrics, four basic parameters need to be known. These parameters are true positive (TP), true negative (TN), false positive (FP) and false negative (FN). The performance metrics calculated using these parameters, Accuracy Equation (2), Precision Equation (3), Recall Equation (4) and F-1 Score Equation (5) are presented below.

$$Accuracy = \frac{\mathrm{TN + TP}}{TN + TP + FP + FN} \tag{2}$$

$$\mathrm{Precision} \ = \frac{\mathrm{TP}}{TP + FP} \tag{3}$$

$$\mathrm{Recall} = \frac{\mathrm{TP}}{TP + FN} \tag{4}$$

$$F1 - Score = 2 * \frac{\mathrm{Precision} \ * \mathrm{Recall}}{Precision \ + \mathrm{Recall}} \tag{5}$$

The analysis in this part of our study utilizes five different deep learning models: DenseNet201, InceptionResNetV2, InceptionV3, NASNetLarge, and Xception. The results of two different datasets, MSID and HYBRID, are compared. The number of epochs was set to 10, the learning rate was 0.0001, and the batch size was 32.

Different deep learning models can perform better with various pooling sizes. For example, the DenseNet201 and NasNetLarge models achieve better results with a pooling size of 7x7, while the InceptionResNetV2 and InceptionV3 models prefer a pooling size of 5x5. The Xception model performs best with a 2x2 pooling size.

DenseNet201 and NasNetLarge: These models use a pooling size of 7x7 to capture features over a larger area. This provides more information and improves classification accuracy [36].

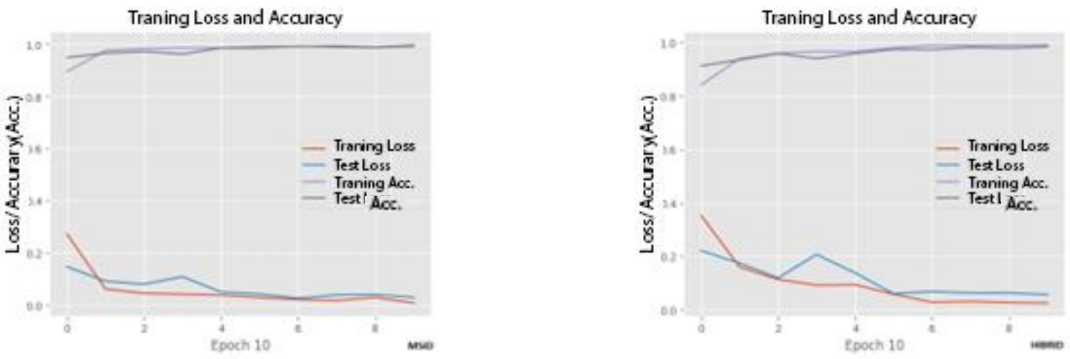
InceptionResNetV2 and InceptionV3: These models prefer a 5x5 pooling size. 5x5 pooling helps to capture finer details in deeper layers of the model, which is especially important in more complex image data [37].

Xception: This model uses a 2x2 pooling size. 2x2 pooling allows the model to run faster and more efficiently by concentrating features in a smaller area. This helps the Xception model run faster and require less computation [37].

These different pooling sizes are optimized for the specific structure of the models and the targeted tasks. The pooling sizes are chosen according to the architecture of each model to achieve the highest performance. The aim of our study is to achieve the highest accuracy in detecting monkeypox across two different datasets using deep learning methods, to analyze the differences between the results, and to uncover significant findings.

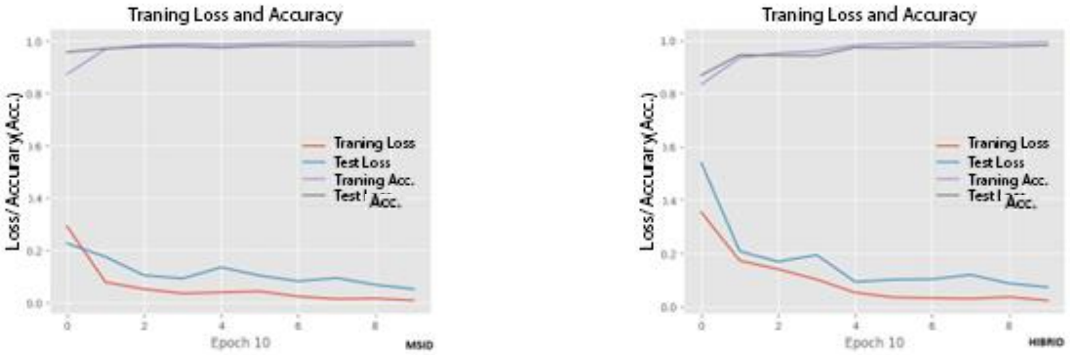
The graphs showing the change in the success rate according to the training and test data are given below. When the graphs are examined, the changes of the models as a result of the analysis in MSID and HYBRID datasets are observed in DenseNet201 Figure 11, InceptionResNetV2 Figure 12, InceptionV3 Figure 13, Xception Figure 14 and NasNetLarge Figure 15.

When the graphs are analyzed, the training and test data of MSID and HIIBRID datasets in all models fluctuate during training, but the most successful results are seen in the MSID dataset. The aim of our study is to obtain the maximum accuracy value of monkeypox disease in two different datasets with deep learning methods, to examine the differences between the values and to reveal important findings. As a result of the studies, the accuracy, precision recall F1-Score, loss values and times of all models are comparatively given i The aim of our study is to achieve the highest accuracy in detecting monkeypox across two different datasets using deep learning methods, to analyze the differences between the results, and to reveal significant findings. The accuracy, precision, recall, F1-Score, loss values, and processing times for all models are comparatively presented in Table 2.



(a) - MSID (b) -HYBRID

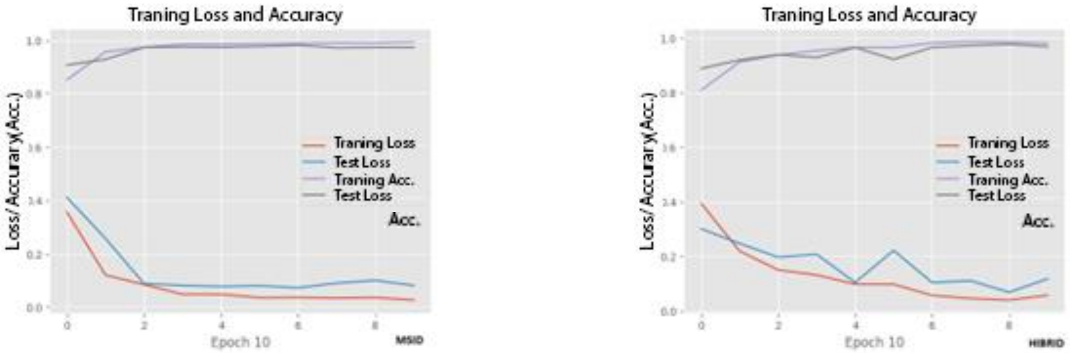
Figure 11. Progress graph of DenseNet201 model on training (a) and (b) datasets



(a) - MSID

(b) -HYBRID

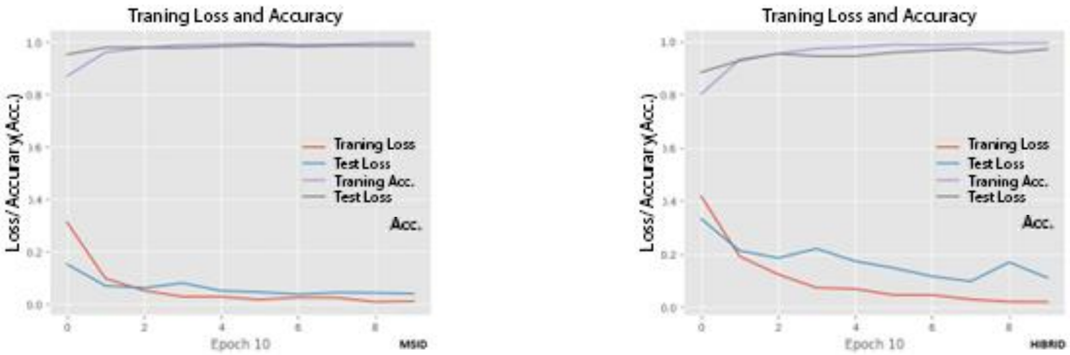
Figure 12. Progress graph of InceptionResNetV2 model on training (a) and (b) datasets



(a) - MSID

(b) -HYBRID

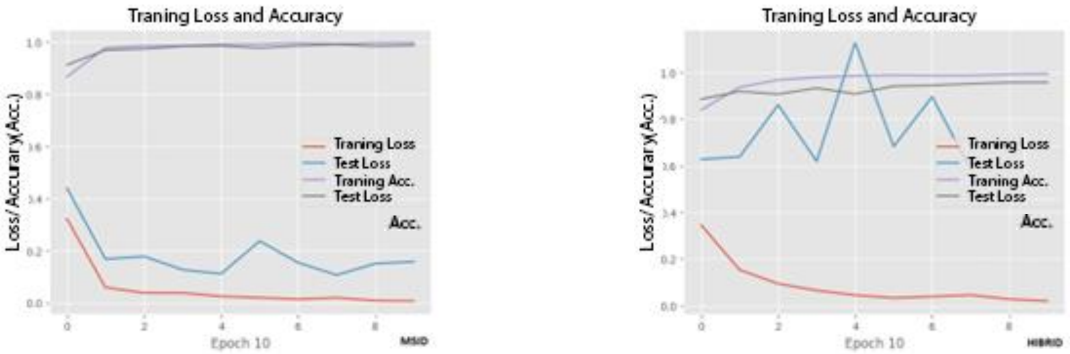
Figure 13. Progress graph of InceptionV3 Model on training (a) and (b) datasets



(a) - MSID

(b) -HYBRID

Figure 14. Progress graph of Xception Model on training (a) and (b) datasets



(a) - MSID

(b) -HYBRID

Figure 15. Progress graph of NasNetLarge Model on training (a) and (b) datasets

Table 2. Test performance evaluation of models

Dataset	Model	Accuracy	Loss	Precision	Recall	F-1 Score	Time
MSID	DenseNet201	0.9933	0.0302	0.99	0.99	0.99	3 h 21 min 21.99 sec
	NasNetLarge	0.9883	0.1579	0.99	0.99	0.99	5 h 33 min 9.44 sec
	Xception	0.9866	0.0404	0.99	0.99	0.99	2 h 10 min 8.79 sec
	InceptionResNetV2	0.9850	0.0531	0.99	0.99	0.99	5 h 13 min 45.6 sec
	InceptionV3	0.9733	0.0815	0.97	0.97	0.97	2 h 29 min 24.16 sec
HYBRID	DenseNet201	0.9851	0.0573	0.99	0.99	0.99	1 h 18 min 56.93 sec
	InceptionResNetV2	0.9812	0.0724	0.98	0.98	0.98	50 min 27.32 sec
	Xception	0.9703	0.1097	0.99	0.99	0.99	55 min 23.87 sec
	InceptionV3	0.9679	0.1167	0.99	0.99	0.99	25 min 25.38 sec
	NasNetLarge	0.9578	0.7200				2 h 3 min 51.28 sec

In line with the evaluations made on the MSID dataset in Table 2, the accuracy rates of the models used are analyzed. The highest accuracy rate was achieved with the DenseNet201 model at 99.33%, whereas the InceptionV3 model recorded the lowest accuracy rate at 97.33%. The accuracy rates of the other models are as follows: 98.50% for InceptionResNetV2, 98.83% for NASNetLarge, and 98.66% for Xception. Looking at the accuracy rates, it is seen that the DenseNet201 model is more successful than the other models. In this case, the ranking of the models according to accuracy rates is as follows: DenseNet201, NasNetLarge, Xception, InceptionResNetV2 and InceptionV3.

In line with the evaluations made on the HYBRID dataset in Table 2, the models used were analyzed according to their accuracy rates. The highest accuracy rate is obtained with the DenseNet201 model with 98.51%, while the lowest accuracy rate belongs to the NasNetLarge model with 95.78%. The accuracy rates of the other models are 98.12% (InceptionResNetV2), 96.79% (InceptionV3) and 97.03% (Xception), respectively. Looking at the accuracy rates, it is seen that the DenseNet201 model is more successful than the other models. In this case, the ranking of the models according to accuracy rates is as follows: DenseNet201, InceptionResNetV2, Xception, InceptionV3 and NasNetLarge.

According to the analysis performed on the MSID dataset shown in Table 2, the recall, precision, loss and f1-score values of the models used were evaluated. The recall, precision and f1-score values of DenseNet201, InceptionResNetV2, NasNetLarge and Xception models were found to be 99% equal and higher than InceptionV3. In the InceptionV3 model, the recall, f1-score and precision values were 97%. When the models are compared, the lowest loss is 0.0302 in DenseNet201 and the highest is 0.1579 in NasNetLarge.

According to the analysis on the HYBRID dataset in Table 2, the recall, loss, f1-score and precision values of the models used were evaluated. The f1-score, precision and recall values of the DenseNet201 model were observed to be 99%. It was found to be higher than InceptionV3, InceptionResNetV2, Xception and NasNetLarge models. The precision, recall and f1-score values of the InceptionResNetV2 model were 98%. In the InceptionV3 and Xception models, the precision, recall and f1-score values were equal at 97%. A decrease in training loss and an increase in training accuracy indicate that the model fits the training data well and that the learning process is effective. Fluctuations in test loss and accuracy indicate that there may be some problems in the generalization ability of the model. In addition, parallel processing capabilities (such as GPU or TPU) negatively affected the training process of the model and f1-score, recall and precision values could not be calculated for the NasNetLarge Model.

More epochs or different regularization techniques can be tried for more stable performance of the model on test data and to reduce fluctuations. Also, parallel processing capabilities (such as GPU or TPU) can speed up the training process of the model and process larger data sets in less time.

5 Conclusions

In this study, maximum accuracy values were obtained with different deep learning models on the detection of monkeypox, one of the epidemic diseases, on two different datasets. Skin lesion images with and without monkeypox disease were classified using different deep learning models. As a result of the study, the DenseNet201 model demonstrated significant success in identifying monkeypox, achieving accuracy rates of 99.33% on the MSID dataset and 98.52% on the HYBRID dataset.

This study offers an important step forward in facilitating the detection of monkeypox disease by integrating technology. In particular, the high accuracy rates achieved by DenseNet201 trained with deep learning models encourage the use of medical imaging systems in monkeypox diagnosis. This technology could help in the early diagnosis and treatment of patients in clinical settings. Moreover, innovative solutions such as mobile applications or remote diagnostic systems can also be effective in containing the spread of disease by providing access to remote areas. In the future, integrating such deep learning models into real-time diagnostic systems and making them widely available could be an important tool in the hands of healthcare professionals and public health experts. With the advancement of technology and the improvement of datasets, it is promising to develop more efficient and reliable methods for early detection of epidemics such as monkeypox.

This study has some limitations. First of all, the limited datasets used and especially the limited representation of monkeypox cases from the general population creates uncertainty about how the model will perform in real-world conditions. Furthermore, the difficulties in collecting and labeling the datasets make it difficult to obtain widely usable and generally valid results.

However, it is critical that future studies utilize more comprehensive and rigorously prepared datasets. This will allow us to more accurately assess the real-world performance of deep learning models in diagnosing rare diseases such as monkeypox. Furthermore, the use of adequately equipped real-time systems is important to reliably integrate the models in clinical applications. In future researchs, it is also recommended to examine the performance of more advanced models, such as Vision Transformer (ViT), a next-generation architecture, on real data sets. Such innovative approaches can further advance the application of deep learning techniques in medical imaging.

Conflict of Interest

The authors declare that they have no conflict of interest.

Similarity rate (iThenticate): %15

References

[1] Tütüncü, D., & Esen, M. Salgın hastalıkların yönetiminde Nesnelerin İnterneti (NİT) kullanımı: COVID-19 örneği. Sağlık Akademisyenleri Dergisi, 8(2), 169-177, 2021.

[2] B. Cerit, Gerçek zamanlı öğrenci takip sistemi ve derin öğrenme yaklaşımı ile maske tespiti. Yüksek Lisans Tezi, Karabük Üniversitesi Mekatronik Mühendisliği Ana Bilim Dalı, Karabük, 2020.

[3] Özer, İ. Türkiye Cumhuriyeti sınırlarını tehdit eden kolera saldırıları ve alınan savunma tedbirleri. Ordu Üniversitesi Sosyal Bilimler Enstitüsü Sosyal Bilimler Araştırmaları Dergisi, 10(1), 201-216, 2020.

[4] Güngör, S., Kaya, M., & Alhajj, R. Renklendirilmiş BT görüntülerinden covid-19 tespiti için derin öğrenmeye dayalı bir yöntem. Computer Science, IDAP-2021: 5th International Artificial Intelligence and Data Processing symposium (Special), 391-399, 2021. https://doi.org/10.53070/bbd.990750.

[5] Robert A. Weinstein, Aysegul Nalca, Anne W. Rimoin, Sina Bavari, Chris A. Whitehouse, reemergence of monkeypox: prevalence, diagnostics, and countermeasures, clinical infectious diseases, 41 (12), 1765–1771, 2005. https://doi.org/10.1086/498155.

[6] World Health Organization. Multi-country monkeypox outbreak in non-endemic countries. https://www.who.int/emergencies/disease-outbreak-news/item/2022-DON385, Accessed 20 Haziran 2022.

[7] K. Adem, Impact of activation functions and number of layers on detection of exudates using circular Hough transform and convolutional neural networks, Expert Systems with Applications, 203, 2022. https://doi.org/10.1016/j.eswa.2022.117583.

[8] Adem, K., Ozguven, M.M. & Altas, Z. A sugar beet leaf disease classification method based on image processing and deep learning. Multimed Tools Appl 82, 12577–12594, 2023. https://doi.org/10.1007/s11042-022-13925-6.

[9] M. Hekim, O. Cömert ve K. Adem, A hybrid model based on the convolutional neural network model and artificial bee colony or particle swarm optimization-based iterative thresholding for the detection of bruised apples. Turkish Journal of Electrical Engineering and Computer Sciences. 28(1), 61-79, 2020. https://doi.org/10.3906/elk-1904-180.

[10] S. Kılıçarslan, K. Adem, and M. Çelik, An overview of the activation functions used in deep learning algorithms, JNRS, 10(3), 75-88, 2021. https://doi.org/10.54187/jnrs.1011739.

[11] E. K. Yılmaz, K. Adem, S. Kılıçarslan, ve H. A. Aydın, Classification of lemon quality using hybrid model based on Stacked AutoEncoder and convolutional neural network. Eur Food Res Technol, 249, 1655–1667, 2023. https://doi.org/10.1007/s00217-023-04245-6.

[12] A. A. Yurdusev, K. Adem, ve M. Hekim, Detection and classification of microcalcifications in mammograms images using difference filter and Yolov4 deep learning model. Biomedical Signal Processing and Control, 80, 2023. https://doi.org/10.1016/j.bspc.2022.104360.

[13] M. Akar, Covıd-19'a karşı insansız hava aracı ile derin öğrenme tabanlı maske tespiti. Yüksek Lisans Tezi, Karamanoğlu Mehmet Bey Üniversitesi Fen Bilimleri Fakültesi, Karaman, 2022.

[14] S. V. Militante ve N. V. Dionisio, Deep learning implementation of facemask and physical distancing detection with alarm systems. 2020 Third International Conference on Vocational Education and Electrical Engineering (ICVEE), 1-5, 2020.

[15] B. Kaya, ve M. Önal, COVID-19 Tespiti için akciğer BT görüntülerinin bölütlenmesi. Avrupa Bilim Ve Teknoloji Dergisi (28), 1296-1303., 2021. https://doi.org/10.31590/ejosat.1015061.

[16] E. Özbay, ve Ö. Altunbey, Derin öğrenme ve sınıflandırma yaklaşımları ile BT görüntülerinden Covid-19 Tespiti. Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi, 12 (2), 211-219, 2020. https://doi.org/10.24012/dumf.812810.

[17] G. H. Christa ve K. M. Sagayam, CNN-based Mask Detection System Using OpenCV and MobileNetV2. 2021 3rd International Conference on Signal Processing and Communication (ICPSC), pp. 115-119, Coimbatore, India, 2021.

[18] M. Shamrat, S. Chakraborty, Md. M. Billah, Md. A. Jubair, M. S. Islam ve R. Ranjan, Face mask detection using convolutional neural network (CNN) to reduce the spread of Covid-19. 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI), pp. 1231-1237, Tirunelveli, India, 2021.

[19] A. A. Jasim, Gerçek Zamanlı Yüz Maskesi Algılama Uyarı Sistemi. Yüksek Lisans Tezi, Kırşehir Ahi Evran Üniversitesi İleri Teknolojiler Ana Bilim Dalı, Kırşehir, 2022.

[20] M. Türkoğlu, K. Hanbay, I. Saraç Sivrikaya, ve D. Hanbay, Derin Evrişimsel Sinir Ağı Kullanılarak Kayısı Hastalıklarının Sınıflandırılması. Bitlis Eren Üniversitesi Fen Bilimleri Dergisi, 9(1), 334-345, 2020. https://doi.org/10.17798/bitlisfen.562101.

[21] K. He, X. Zhang, S. Ren, ve J. Sun, Delving deep into rectifiers: surpassing human-level performance on ImageNet Classification. 2015 IEEE International Conference on Computer Vision (ICCV), pp. 1026-1034, Santiago, Chile, 2015.

[22] Y. Alqadiri, Face mask detection using deep learning methods. Yüksek Lisans Tezi, Bahçeşehir Üniversitesi Bilgisayar Mühendisliği Ana Bilim Dalı, İstanbul, 2022.

[23] A. L. Maas, A. Y. Hannun, ve A. Y. Ng, Rectifier nonlinearities improve neural network acoustic models. Proceedings of the 30th International Conference on Machine Learning, 28 (3), Atlanta GA, USA, 2013.

[24] I. Goodfellow, Y. Bengio, A. Courville, ve Y. Bengio, Deep Learning (Adaptive Computation and Machine Learning series). MIT Press, 2016.

[25] A. Atcılı, Yapay Sinir Ağlarında Kullanılan Optimizasyon Algoritmaları. Machine Learning Turkiye. https://medium.com/machine-learning-t%C3%BCrkiye/yapay-sinir-a%C4%9Flar%C4%B1nda-kullan%C4%B1lan-optimizasyon-algoritmalar%C4%B1-3e87cd738cb5, Accessed 30 Aralık 2022.

[26] H. Çetiner, Yaprak hastalıklarının sınıflandırılabilmesi için önceden eğitilmiş ağ tabanlı derin ağ modeli. Adıyaman Üniversitesi Mühendislik Bilimleri Dergisi, 8(15), 442-456, 2021. https://doi.org/10.54365/adyumbd.988049.

[27] H. Bingol, Classification of OME with eardrum otoendoscopic images using hybrid-based deep models, NCA, and Gaussian method. Traitement du Signal, 39(4), 1295-1302, 2022. https://doi.org/10.18280/ts.390422.

[28] X. Yu, N. Zeng, S. Liu, ve Y. D. Zhang, Utilization of DenseNet201 for diagnosis of breast abnormality. Machine Vision and Applications 30, 1135–1144, 2019. https://doi.org/10.1007/s00138-019-01042-8.

[29] B. Zoph, V. Vasudevan, J. Shlens, ve Q. V. Le, Learning transferable architectures for scalable image recognition. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8697-8710, Salt Lake City, UT, USA, 2018.

[30] F. Bozkurt, Derin öğrenme tekniklerini kullanarak akciğer X-Ray görüntülerinden COVID-19 tespiti. Avrupa Bilim Ve Teknoloji Dergisi, 24, 149-156, 2021. https://doi.org/10.31590/ejosat.898385.

[31] M. M. A. Ashames, Derin öğrenme yöntemleri kullanılarak osteoporozun belirlenmesi. Yüksek Lisans Tezi, Konya Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü, Konya, 2020.

[32] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke ve A. Rabinovich, Going deeper with convolutions. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1-9, Boston, MA, USA, 2015.

[33] C. Szegedy, S. Ioffe, V. Vanhoucke, A. Alemi. Inception-v4, Inception-ResNet and the impact of residual connections on learning. Proceedings of the AAAI Conference on Artificial Intelligence, 31(1), Phoenix, Arizona, 2016.

[34] Keras.io. (2022). Keras documentation: The Model class. https://keras.io/api/models/model/, Accessed 29 December 2022.

[35] Ö. Doğan, CNN (Convolutional Neural Networks) Nedir?, https://teknoloji.org/cnn-convolutional-neural-networks-nedir, Accessed 28 December 2022

[36] C. Szegedy, S. Ioffe, V. Vanhoucke, J. Shlens. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2818-2826, Las Vegas, NV, USA, 2016.

[37] A. Dash, P. Sethy, Statistical analysis and comparison of deep convolutional neural network models for the identification and classification of maize leaf diseases. Multimedia Tools and Applications, 1-14, 2024. https://doi.org/10.1007/s11042-024-18481-9.