NEAR EAST UNIVERSITY

INSTITUTE OF GRADUATE STUDIES

DEPARTMENT OF COMPUTER ENGINEERING

USING OF CONVENTIONAL NEURAL NETWORKS TO DIAGNOSE SCABIES BY DERMOSCOPY EVALUATION OF OPINION OF TEACHERS

M.Sc. THESIS

Husam ZENDAH

Nicosia

June 2022

HUSAM ZENDAH

USING OF CONVENTIONAL NEURAL NETWORK TO DIAGNOSE SCABIES BY DERMOSCOPY.

MASTER THESIS 2022

NEAR EAST UNIVERSITY

INSTITUTE OF GRADUATE STUDIES

DEPARTMENT OF COMPUTER ENGINEERING

USING OF CONVENTIONAL NEURAL NETWORKS TO DIAGNOSE SCABIES BY DERMOSCOPY.

M.Sc. THESIS

Husam ZENDAH

Supervisor
Assoc. Prof. Dr. Kamil Dimililer

Nicosia

June 2022

Approval

We certify that we have read the thesis submitted by Husam ZENDAH titled "**USING OF CONVENTIONAL NEURAL NETWORK TO DIAGNOSE SCABIES BY DERMOSCOPY.**" and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Educational Sciences.

Examining Committee	Name-Surname	Signature
Head of the Committee:	ASSOC. PROF. DR. YÖNEY KIRSAL EVER	,,,,...........................
Committee Member*:	ASSOC. PROF. DR. BORAN ŞEKEROĞLU	....
Supervisor:	ASSOC. PROF. DR. KAMİL DİMİLİLER	....

Approved by the Head of the Department

...../...../2022

....

Prof.Dr. Rahib H.Abiyev

Head of Department

Approved by the Institute of Graduate Studies

...../...../2022

Prof. Dr. Kemal Hüsnü Can Başer

Head of the Institute

Declaration

I hereby declare that all information, documents, analysis, and results in this thesis have been collected and presented according to the academic rules and ethical guidelines of the Institute of Graduate Studies, Near East University. I also declare that as required by these rules and conduct, I have fully cited and referenced information and data that are not original to this study.

Husam ZENDAH

27/06/2022

ACKNOWLEDGEMENTS

My deepest gratitude goes to my supervisor Assoc. Prof. Dr. Kamil Dimililer, for his exemplary guidance, patience, and support during this thesis. This thesis would not have been possible without his help and vast wisdom. I have to say a big thank you to my parents for their prayers, financial support, and everything. I will not forget my friends that care about me so much and always ask about my thesis and how is going.

Husam ZENDAH

Abstract

USING OF CONVENTIONAL NEURAL NETWORK

TO DIAGNOSE SCABIES BY DERMOSCOPY

ZENDAH, Husam

MSc, Department of Computer Engineering

June 2022, 87 Pages

In medicine, particularly dermatology, artificial intelligence has made major advances. The usage of different networks for scabies detection, including such deep learning, has proven to be highly advantageous in terms of overall performance. The goal of this work, however, is to conduct an analytical inquiry for the early diagnosis of scabies using a deep learning system trained using a comparison between VGG-16 and a traditional model. The objectives of this study were the investigation of deep learning, the collection of quantitative data, the use of VGG-16 and a traditional model of both testing and training, as well as the interpretation of results. The method involved reviewing a dataset acquired again for study and using a higher processing machine that boosts the effectiveness of results. VGG-16 and a traditional model have been trained and tested on same two different categories. The VGG-16 and a traditional model performed exceptionally well in the research, with an acceptable accuracy of 97.67 percent, and 88.37 percent respectively. Aside from developments in hardware technology and processing, new categorization approaches based on deep learning have raised the relevance of dermatological applications. Many difficulties may be solved through these programs, such as limited access owing to distance, physical impairment, a lack of dermatologists, jobs, scheduling, and so on. They also assist doctors in making objective and timely diagnostic judgments. As a result, therapies may be administered on time, which is especially critical for life-threatening skin illnesses.

Key Words: CNN; deep learning; transfer learning; scabies; Dermoscopy images; diagnosis.

Özet

USING OF CONVENTIONAL NEURAL NETWORK

TO DIAGNOSE SCABIES BY DERMOSCOPY

ZENDAH, Husam

MSc, Department of Computer Engineering

June 2022, 87 Pages

Tıpta, özellikle dermatolojide yapay zeka büyük ilerlemeler kaydetti. Bu tür derin öğrenme dahil, uyuz tespiti için farklı ağların kullanılmasının, genel performans açısından oldukça avantajlı olduğu kanıtlanmıştır. Ancak bu çalışmanın amacı, VGG-16 ve geleneksel bir model arasında bir karşılaştırma kullanılarak eğitilmiş bir derin öğrenme sistemi kullanarak uyuz hastalığının erken teşhisi için analitik bir araştırma yapmaktır. Bu çalışmanın amaçları, derin öğrenmenin araştırılması, nicel verilerin toplanması, VGG-16'nın kullanımı ve hem test hem de eğitim için geleneksel bir model ve sonuçların yorumlanmasıydı. Yöntem, çalışma için tekrar elde edilen bir veri kümesinin gözden geçirilmesini ve sonuçların etkinliğini artıran daha yüksek bir işleme makinesinin kullanılmasını içeriyordu. VGG-16 ve geleneksel bir model, aynı iki farklı kategoride eğitilmiş ve test edilmiştir. VGG-16 ve geleneksel bir model, sırasıyla yüzde 97,67 ve yüzde 88,37 kabul edilebilir bir doğrulukla araştırmada son derece iyi performans gösterdi. Donanım teknolojisi ve işlemedeki gelişmelerin yanı sıra, derin öğrenmeye dayalı yeni kategorizasyon yaklaşımları dermatolojik uygulamaların önemini artırmıştır. Mesafe nedeniyle sınırlı erişim, fiziksel bozukluk, dermatolog eksikliği, iş, zamanlama vb. gibi birçok zorluk bu programlar aracılığıyla çözülebilir. Ayrıca doktorlara objektif ve zamanında teşhis kararları vermelerinde yardımcı olurlar. Sonuç olarak, tedaviler zamanında uygulanabilir ve bu özellikle yaşamı tehdit eden cilt hastalıkları için gerçekten kritiktir.

Anahtar Kelimeler: CNN; derin öğrenme; transfer öğrenme; uyuz; dermoskopi görüntüleri; Teşhis.

Table of Contents

CHAPTER IV

CHAPTER V

List of Tables

List of Figures

List of Abbreviations

AI	Artificial Intelligence
AUC	Area Under Curve
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CT	Computer Tomography
DL	Deep Learning
GPU	Graphical Processing Unit
ML	Machine Learning
MLP	Multilayered Perceptron
ReLU	Rectified Linear Unit
RGB	Red, Green, Blue
TL	Transfer Learning
TPU	Tensor Processing Unit

CHAPTER I
Introduction

Overview

The skin is the biggest organ in the human body and serves various functions. Because the skin is exposed to the outside world, illnesses and infections are more common. A lesion area is an infectious patch on the skin. Despite technological improvements and advances in the domains of medicine, the number of individuals affected by illnesses [1] continues to rise. Worse, most of these diseases remain undiscovered until they are in their latter stages, giving them fewer odds of survival. Many lives have been lost as a result of a lack of procedures to raise public awareness, for a variety of causes. Some of them cannot afford to contact a doctor due to financial constraints, others lack the time to do so, and still, others are unaware of the extent of their influence. To preserve the balance between expansion in the medical sector, care should be made at this time to ensure that people are well informed of what is ahead of them. Deep Learning is an Artificial Intelligence area in which a computer algorithm analyses raw data and learns the discriminating features needed to uncover hidden patterns in it. During the preceding decade, there were considerable advances in the ability of DL-based programs to evaluate various types of data, notably photos. The most common DL model can be trained by supervised methods, in which datasets comprise inputs (for example, dermoscopic images of skin disorders) and direct contractual labels (for example, diagnosis or skin disease categories such as 'Normal' or 'Abnormal'). Recent developments in image categorization and object recognition [4] can considerably assist healthcare and medicine, particularly those medical fields where diagnoses are predominantly reliant on pathology, radiography, optometry, and dermatitis for the identification of morphologic alterations, among others. Digital pictures are taken in such medical areas and fed into DL algorithms for Computer-Aided Diagnosis (CAD). As a result of these factors, Constructing image analysis devices becomes a significant area of study. In the hands of unskilled dermatologists, dermoscopy has been shown to improve the diagnosis accuracy of skin disorders. This research examines common skin scabies conditions with distinctive symptoms that may be exploited as image recognition objects.

Aim of Study

In medicine, artificial intelligence is commonly employed. Aside from developments in hardware technology and processing, new categorization approaches based on deep learning have raised the relevance of dermatological applications. Many difficulties may be solved through these programs, such as limited access owing to distance, physical impairment, a lack of dermatologists, jobs, scheduling, and so on. They also assist doctors in making objective and timely diagnostic judgments. As a result, therapies may be administered on time, which is especially critical for life-threatening skin illnesses. Costs can also be decreased. Furthermore, the software is free or less expensive than an in-person medical consultation. Because they can prevent cost increases by (1) recognizing illnesses at an early stage (which decreases treatment costs because the condition is less progressed), and (2) requiring fewer doctor visits (because most primary care is delivered through consultations).

Significance of Study

Artificial intelligence has seen tremendous progress in the realm of medical-assisted diagnosis, and deep learning technology is crucial in medical picture identification. They have, however, mostly been designed to diagnose one or a few particular skin disorders. As a result, one of the outstanding concerns is how to develop deep learning-based apps to diagnose additional skin illnesses. Furthermore, these apps have not been evaluated using widely available benchmark datasets. As a result, comparing their achievements based on the values stated in the papers is meaningless. Furthermore, specialists, medical physicians, and patients remain concerned about the diagnostic accuracy and safety of applications designed for skin disease diagnostics. Because the majority of these uses have not been verified through clinical trials. Generally, photos from libraries are used for training and testing. They do not, however, resemble lesions encountered by dermatologists in practice. Deep learning-based algorithms have increased classifier confidence, given quicker computation rates, and better outcomes. Deep CNNs will deliver more professional information in the future due to technological progress, inter-disciplinary effort, and database enhancement with greater picture volume and quality. As a result of developments in deep learning-based methodologies, ubiquitous application utilization is projected to play a key role in dermatology. However, further research is needed in this sector.

Thesis Structure

The content of the thesis is divided into the following sections:

Chapter 1: Introduction

This chapter, describes the introduction of the research, the aim of the study, the significance of the study, and the Thesis structure.

Chapter 2: Literature Reviews

This chapter, reviews some research some studies have done before on the same problem.

Chapter 3: Methodology

This chapter will do an overview of Convolutional Neural networks.

Chapter 4: Simulation and Results

This chapter will do an overview of Convolutional Neural networks, VGG16, and Transfer Learning.

Chapter 5: Conclusion and Recommendation

This chapter describes the summaries of the experiments and recommendations for future work in the area.

CHAPTER II

Literature Review

Introduction

This chapter will go through what individuals have done in the past about this project. Then we'll go through in-depth what's being done to create this unique system.

Literature Review

In recent years, technology has advanced dramatically in all aspects of life. But first, let's take a look at how artificial intelligence has evolved in recent years.

This paper offers an end-to-end structure for human skin recognition by adding recurrent neural layers with FCNs. RNN layers are utilized to express the semantic spatial relationships between image pixels. Experiment results show the proposed FCN and RNN strategy outperforms existing techniques both on the COMPAQ and ECU body datasets. RNN layers boost the efficiency of skin technical skills and experience in difficult situations. (Haiqiang, et al, 2017)

In this paper, designers look at automated skin lesion assessment, namely melanoma identification, and semantic segmentation. They do this by utilizing deep learning algorithms to categorize publicly available dermoscopic images. In this research, he describes our attempts to create a system that is easy to use and depends on deep learning for skin lesion categorization, resulting in an improved melanoma detection system. Deep convolutional neural network architecture is initially created over raw images for categorization. There are additional 166-D histogram distributions, edge histograms, and other hand-coded features, as well as Multiscale. To extract local binary color patterns from pictures, a random forest classifier is utilized. The final classification result is the average of the outputs from the two specified classifiers. The classification job has 80.3 percent accuracy, an AUC of 0.69, with an accuracy of 0.81 For segmentation, With a Dice coefficient of 73.5 percent, they use a convolutional-deconvolutional architecture. (Singh, et al, 2018)

Convolutional Neural Networks (CNNs) have the potential to help physicians with diagnosis and therapy. This work was written to assist dermatologists by offering foundational knowledge on deep learning and CNNs, as well as (ii) applications of CNNs for skin disease categorization. Furthermore, while CNN-based approaches offer great potential for automated diagnosis, further study and novel techniques in image processing and pattern recognition are needed to enable more accurate detection of dermatological illnesses. These two types of CNN applications in dermatology have been addressed in this work: illness classification based on medical pictures (for example, dermoscopy and pathological images); (ii) and disease classification based on digital photos. As a result, this paper makes two significant contributions: First, the fundamental ideas of deep learning and CNNs are explained, which will aid dermatologists in understanding and implementing CNN-based automated procedures. Second, cutting-edge applications for lesion categorization from medical pictures and color photography are given. The downsides or limits of these applications are also discussed. Furthermore, this report suggests a scarcity of desktop programs designed for dermatological illnesses other than skin cancer. (Göçeri and colleagues, 2020)

A generally established clinical guideline for melanoma screening is the observation of lesion changes in the near term. When a melanocytic lesion changes significantly after three months, it is removed to rule out melanoma. However, whether to alter or not to change is largely influenced by the experience and prejudice of individual physicians, which is subjective. For the first time, an unique deep learning-based technique for automatically detecting short-term lesion alterations in melanoma screenings is developed in this work. The lesions change detection problem is characterized as one that compares the similarity between two dermoscopy images recorded for a lesion in a short amount of time, and new Siamese institutional characteristics deep network is provided to deliver the decision: altered (namely not the same) or unchanged (i.e. similar enough). In contrast to deep convolutional features, the Siamese framework presents a novel structure, Generates a significant Prediction Process, to recover the global properties of lesion images. . A segmented loss (SegLoss) is then constructed and implemented as a smoothing filter in the proposed network to mimic the judgment process of clinicians, who typically focus mostly on areas with specific characteristics when comparing a group of lesion images. To evaluate the proposed technique, a dataset of 1,000 pairings of lesion images gathered in a short period at a clinical cancer center was created. Experiment results on this unique massive dataset reveal that the proposed technique is effective for identifying objective short-term lesions change Melanoma screening. (Zhang, et al, 2020)

(Rodrigues et al, 2020) Early identification It is crucial to detect carcinoma, one of the worst types of cancer. The use of Deep Neural Networks (CNNs) is now the primary area of inquiry for the automated identification of this type of sickness. However, the majority of previous works were created based on transferring General-purpose architectures that were applied to the field of skin lesions, resulting in inflexibility and excessive processing costs. This paper provides a one-of-a-kind architecture that makes use of cutting-edge CNN methods. Their architecture was designed and trained from the ground up with melanoma in mind, employing Aggregated Transformations and the Squeeze-and-Excite approach in a residual block. His findings show that a solution like this can compete with big-state melanoma detection methods. With a quarter of the weight of previous efforts, this approach is adaptable and delivers low processing costs for real-world in-place applications. (Rodrigues et al, 2020)

The authors present a novel dermoscopic image detection and classification method proposed based on a Deep Neural Network (CNN). in this research. Five disorders were first investigated, and this number may be raised in the future. The Classified approach specifies the nature of the skin disease: The Softmax classifier is the network's last layer, producing an actual likelihood for each label. The image analysis unit, as well as the classification unit, are the two main components of the architecture. A first training yields an approximate output accuracy of 70 %. (Rathod et al, 2018)

This work introduces a CNN with enhanced U-Net architecture, and the ResNet architecture is employed in the encoding part. The suggested technique outperforms existing CNN designs in terms of stability and efficiency for the supplied dataset. Accuracy rates of 92 percent for training data and 89 percent for testing data have been reached. The dermatology picture collection contains photographs of three different forms of skin injury, one of which is skin cancer. (Iranpoor and colleagues, 2020)

A method based on machine learning for diagnosing seven kinds of skin disorders using a neural network using convolutions (CNN) is described here. Transfer learning, in conjunction with CNN, was utilized to increase classification precision on the 2018 International Skin Imaging Collaboration dataset. It was discovered that applying transfer learning instead of simply CNN increased accuracy by 11%. When compared to previous efforts, the performance of this suggested technique seems promising. Actinic keratoses and intraepithelial carcinoma/disease Buckley's, basal cell cancer (bcc), innocuous keratoses-like illnesses (solar lentigines/seborrheic keratoses and endophyte keratoses, dermatofibroma, and melanoma were discovered (angiomas, angiokeratomas, pyogenic granulomas, and hemorrhage. To improve accuracy, it utilized VGG16 classifiers, which are often used in CNN and Transfer Learning. In this study, 1137 photos are used for training and 197 photographs are used for testing. In this case, it can be concluded that transfer learning surpasses CNN architecture by 11%. (Guha et al, 2019)

Melanoma is the most lethal kind of skin cancer. Convolutional Neural Networks, in particular, are deep learning approaches (CNNs), that have been utilized for decades in computer vision applications. The activation functions in our model are simply modified to tangent and ReLU functions; all other hyperparameters remain unaltered. untouched. Designers train all networks with a 10015 training dataset of skin photographs and evaluate them with a test set of 1500 images. In this network, Hidden layers of APL units outperform Tangent functions and ReLUs. Furthermore, after around 65 epochs, our APL-based network achieves a peak training accuracy of about 98 percent and then remains steady. A similar network trained using ReLUs and slope functions has a training accuracy of less than 96 percent and becomes stable after 150 and 200 epochs, respectively. (Namozov et al2018)

The majority of previous studies in cases of skin illness employed neural networks using convolutions (CNN) with conventional loss functions, which limited the model's ability to discover distinguishing characteristics from skin pictures. To solve the aforementioned issue, they suggested a novel architecture based on fine-tuning layers of the ResNet152 and InceptionResNet- A triplet error rate is used in V2 models. First, In the proposed approach, researchers train the embedding based on input images deep CNN into Euclidean space InceptionResNet-V2 and ResNet152 models. Second, To learn the discriminative properties of skin disease photographs, scientists apply the loss function for triplets to compute the L-2 distance between matching pictures in euclidean space. The photos of human faces with skin diseases utilized in the suggested framework were obtained from a hospital in Wuhan, China. Because of the importance of four major medical diagnosis approaches such as ABCD guidelines, Pattern analysis, Menzies technique, and 7-Point Checklist in accurately analyzing melanoma skin cancer. Using the triplet loss function, Researchers created a revolutionary deep CNN-based skin disease classification model. The dataset for the experiment comprises four types of skin diseases: Acne symptoms include pimples, blackheads, dark patches, and spots. To evaluate the approach, 12000 input photos were utilized for instruction, 2,000 for testing, and 10 percent for administration of the data for training the validation set. In terms of skin disease categorization, the strategy beats current best practices. The proposed technique may also be used to classify diseases in other contexts. Because biological taxonomy organizes dormant pictures, the suggested method's performance may be enhanced by developing a dataset with the assistance of a dermatologist to visually structure taxonomy. (Ahmad et al, 2020)

An intelligent diagnostic multi-class skin lesion detection system categorization is suggested. The suggested technique employs a hybrid approach, namely An error-correcting outputs code with a deep deep convolutional neural network (ECOC) support vector machine. The suggested approach is intended to categorize skin lesion images Healthily, acne, eczema, benign, or melanoma are the five classifications. Experiments were carried out on 9,144 photos gathered from various sources. The features were AlexNET, a pre-trained neural network, was used to retrieve the CNN model. The ECOC SVM classifier was utilized for classification. The total accuracy attained with ECOC SVM is 86.21 percent. To minimize overfitting, cross-validation of the tenfold procedure was applied. The results show that characteristics derived from the convolutional neural network can improve the classification performance of various skin lesions. (Hameed et al, 2018)

Neural networks have improved significantly in various aspects of artificial intelligence. Convolutional neural networks have a distinct design that enables training on high-dimensional input such as pictures while utilizing limited computational resources. It does this by using three architectural concepts to assure a certain degree of invariance in shift and distortion: shared weights, local receptive field, and occasionally spatial or temporal subsampling. The authors now suggest a design based on Convolutional Neural Networks for detecting Leprosy lesions. To train the network, the authors employ DermnetNz datasets together with online scraped photos to reach the best accuracy of 91.6 percent on a dataset divided into 60% training images, 20% are still not. Cross pictures for validation, and 20% for testing. Because there were only 120 training and validation examples in the training dataset, the picture samples were randomly geometrically modified inside each training cycle to enhance the usefulness of the gathered photos. (Baweja et al, 2016)

The goal of this research is to better understand the effectiveness of approaches based on convolutional neural networks (CNNs). They begin by creating two types of databases on skin disorders using photographs from the Internet: (a) Skin -10, a dataset of 10,218 pictures of 10 major skin disease classifications; (b) Skin -100, a larger dataset with There are 19,807 images representing 100 distinct skin disease classes. These datasets serve as the foundation for our benchmarking. numerous SOTA CNN models and show that skin -100 has significantly lower accuracy than skin -10. Then, by employing an ensemble method based on several CNN models, scientists get the greatest accuracy of 79.01 percent for Skin -10 and 53.54 percent for Skin -100. It also demonstrates a unique receptors system approach that makes use of the Skin -10 dataset and bounding boxes. According to our findings, object recognition can help improve the accuracy of various skin disease classifications. (Xin He et al,2019)

Clinical photos recorded from Everyday devices are prone to visual corruption such as motion blur and noise, which may easily deceive the automated system. As a result, the goal of this article is to develop a resilient and transportable deep convolutional neural network (DNN) to discriminate HZ from other skin illnesses using user-submitted photos. to boost resilience while keeping They provide knowledge distillation from ensembles at a low computational cost (KDE-CT), in which a student network gradually learns from a more powerful teacher network. Researchers developed skin disease information for HZ diagnosis and tested its resistance against 75 distinct types of corruption. Thirteen different DNNs were evaluated on both clean and damaged images. The experiment findings show that the suggested KDE-CT significantly improves corruption resilience when compared to existing techniques. The trained MobileNetV3-Small beat the DNN ensemble (93.5 percent overall accuracy, with a mean corruption error of 67.6) with less (549x fewer multiply-and-accumulate operations), making it excellent for diagnosing migratory skin lesions. (Seunghyeok et al, 2021)

CNN, which stands for convolutional neural network, was chosen as a neural network of choice. Previously, DNN, or deep neural network, was employed for detecting work. Hand dermatitis, eczema, subacute eczema, lichen simplex, skin irritation, and ulcers are now classified. This research combines image processing techniques with machine learning. Where image processing has produced the graphic used by CNN to organize the lessons The above-mentioned five sorts of skin gives are included in the planning information. Researchers achieved 73 percent accuracy using our technique on the dormant dataset of 500 photos of various conditions. If the remaining enhancements are accomplished using a bigger piece of the sample, this will be a tremendous feat. (Rimi et al, 2020)

Existing melanoma evaluation models evaluate skin lesions using either pattern analysis approaches or seven-point checklist criteria. This research presents a pattern analysis technique for melanoma detection that incorporates a seven-point checklist and uses a convolutional neural network to extract lesion details properly. The value of features learned automatically derived from dermoscopic pictures has been created, achieved, and assessed using stacked layers of convolution filters. The constructed Cnn models (CNNs) with numerous inputs were employed in clinical and dermoscopic studies. pictures as input, with a distinct Each picture type has its feature extraction model. Both models' characteristics are concatenated to interpret and ultimate prediction of lesion kind The sum of the weights for predicated lesions calculated based on seven-point checklist criteria is then passed into a threshold model, which are Atypical Pigment Network, Blue Whitish Vei, Vascular Structure, Irregular Pigmentation (PIG), Irregular Streaks, Irregular Dots and Globules, and Regression Structures are all examples of irregular pigmentation to determine whether or not the image is normal (melanoma or non-melanoma). A dataset of 2000 dermoscopic pictures is used to evaluate the performance of the created algorithms early results of the suggested technique demonstrate a convincing and promising capability for lesion detection and automatic melanoma diagnosis utilizing dermoscopy images. (Alzahrani et al,2019)

In this study, they offer a system that leverages contemporary deep learning approaches to identify melanoma by classifying skin lesions. Dermoscopy is the name given to a specialized way of imaging the skin at high resolution. It decreases skin surface reflectivity and helps physicians to examine deeper features. Dermoscopy is used to diagnose with an accuracy rate of about 75-84 percent. However, the success rate is dramatically reduced as a result of non-specialist doctors' diagnostic results. A melanoma detection method based on skin pictures is provided. The suggested technique was evaluated using one of the biggest melanoma datasets available. The dataset comprises around 350 pictures for 900 training tests. To achieve great accuracy, deep learning algorithms require a huge number of photos. Despite this, the system was effectively trained with a small number of data. Despite this flaw, the suggested system was able to achieve 70 percent accuracy, 0.4 average precision, and a 0.71 f-score after a series of trials. In the research that used the identical data set and yielded the best results, an accuracy of 85 percent and an average accuracy of 0.64 were attained. The given study demonstrated that satisfactory results may be obtained without the use of complicated data processing and picture segmentation procedures. (Alper Arik et al, 2017)

CHAPTER III
Methodology

Design

The study's design is a computer program of a diagnostic algorithm for scabies cases based on photographs. The neural network using convolutions (CNN) deep learning technique is employed A Tensor processing unit was used in the implementation and serves as the equipment that performs the code in feature extraction, deep learning is used to reduce training time and avoid overfitting, and increasing algorithm accuracy. The model implementation is divided into four major stages, which are as follows:

A) Input phase

B) Pre-processing phase

C) Training phase

D) Output phase

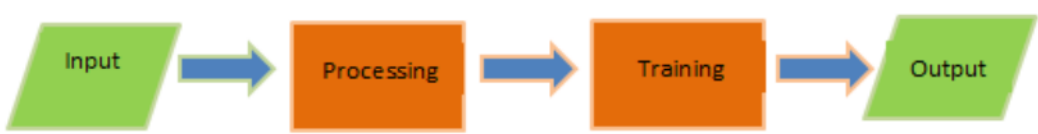

Figure 1 Map of the Designing Phase in Blocks.

Input phase: The model's input is a red, green, and blue (RGB) scabies picture in jpg format. This is the algorithm's initial step. After the scabies photos are fed into the model, the following stage, pre-processing, will resume.

Pre-processing phase: That's the algorithm's second phase when the input scabies photos are processed. examined to see whether they were RGB form, If they are not, they are transformed to RGB form, the scabies photographs are scaled to 180 × 180 pixels are divided each input image to normalize Pixels are created by multiplying pixel by 255 vary instead of The original, between 0 until 1 picture pixel span was 0 – 225. The preliminary stage enables the algorithm to quickly learn and extract crucial picture information.

Training phase: The third step of the model is training. The pre-processed scabies photos are fed into the neural network using the convolutions input layer, then go to the unseen layers containing characteristics that are learned and retrieved, and finally to the output layer through the result of the previously hidden layer.

The output phase is the model's final and final stage. The model's ultimate conclusion is made at this network level (output layer) by creating two output classes: Normal and Abnormal. This choice is made by The deep neural network has the last layer, Which is in the shape of a likelihood. When a Normal The algorithm's class has a greater coefficient of determination choice is Normal; when an Abnormal likelihood value for the class is greater, the algorithm's decision is Abnormal.

System Architecture

An input layer is the foundation of a convolutional neural network, hidden units, as well as an output layer. The inputs and results of any intermediate layers in a flow neural network are obscured by the input signal and final convolution. Convolutional layers are included in the depths units of a deep neural network. Typically, this consists of a layer that performs a linear combination of the convolutional network and the layer's input matrix. Each hidden layer is made up of a group of several neurons, each of which is entirely linked to every neuron in the neurons in the previous layers, and cells in a thin layer act totally They operate separately and therefore do not share any relationships. As the convolution operation slides all along the input sequence for the layer, a feature map is created, which subsequently adds to the input of a subsequent stage. The last fully connected level is referred to as the "output layer," and it provides the classification contexts and the class scores.

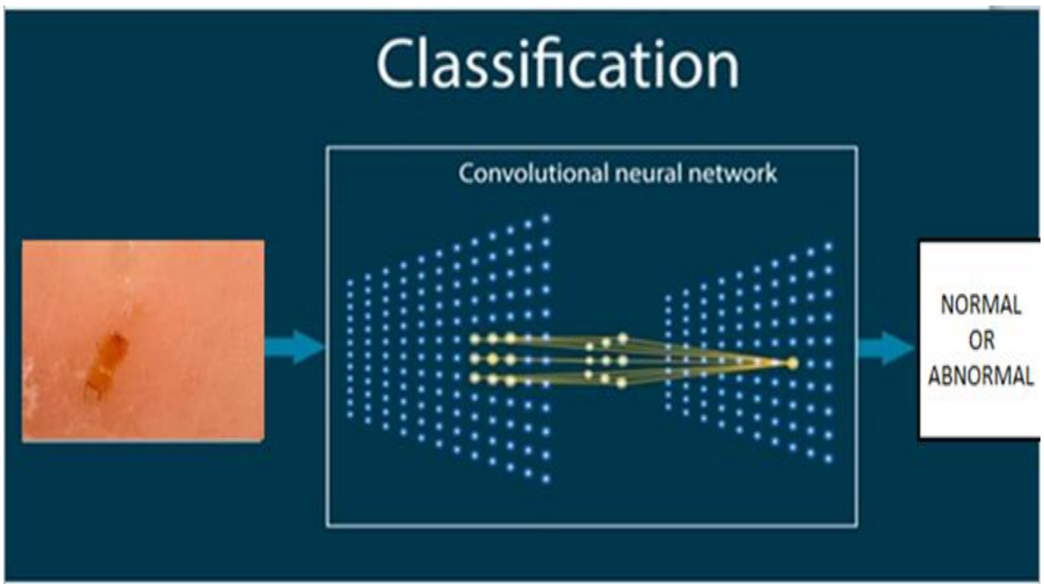

Figure 2 Block Diagram of CNN Model.

Dataset

The rising prevalence of scabies has lately prompted the development of computer-aided diagnostic tools for dermoscopic picture categorization. The TRIPOLI MEDICAL CENTER dataset was created for research and benchmarking purposes, to promote comparative studies on dermoscopic image segmentation and classification methods. TRIPOLI MEDICAL CENTER is a dermoscopic picture database collected from TRIPOLI MEDICAL CENTER's Dermatology Service in Tripoli, Libya. The dermoscopic pictures were acquired under identical conditions at the Dermatology

Service of Hospital using the Dermoscpoy equipment at a magnification of 20x. They have an 8-bit RGB color scheme. graphics with a 768x560 pixel resolution. This picture database comprises 138 dermoscopic photos of scabies lesions, comprising 74 Abnormal images and 64 Normal images. The TRIPOLI MEDICAL CENTER All medical annotations are included in the database photos, including lesion medical segmentation. Each parameter was evaluated by an experienced dermatologist.

Figure 3 Scabies/Abnormal /Normal Dermoscopy images.

Data Preparation

We'll spend a lot of time on data pretreatment techniques that are often employed in image processing. This is because preprocessing consumes around 50–80 percent of the time in most deep learning projects, and it includes both pixel value scaling and the usage use of digital image augmentation methods during both training and testing assessment of the model. Rather than evaluating a wide variety of possibilities, examine the forms of data preprocessing, train-time enhancement, and test-time enhancement. Employed by cutting-edge models that obtain notable results on a difficult computer vision dataset, specifically the Large Scale Visual Recognition Competition, which utilizes the ImageNet dataset Training set should be supplemented most likely include rescaling at random, vertical flips, luminance, contrast, and color perturbations, in addition to random cropping. Because the photographs in the training dataset were of varying sizes, they would have to be resized before they could be used as model input. It should be noted that the network demands input pictures to have the form 224224, which is obtained by training augmentation. ImageNet is made up of photos with varying resolutions; however, our algorithm requires a consistent input dimensionality. As a result, we downsized the photos to a fixed resolution of 64 64. Deep Convolutional Neural Networks for Image Net Classification After that, Every pixel has its mean pixel value removed, a process known as centering. It is assumed that this was done per channel: the average pixel values were calculated from the training dataset for each of the color pictures' Channels in red, green, and blue. We didn't do any extra pre-processing on the photographs.

Reorganize the data

Each picture is 64X64 pixels in size, for a total of 784 pixels. So the output layer has two outputs, the hidden layer has seven hundred and eighty-four neurons, and the input layer has seven hundred and eighty-four inputs. After that, the dataset is transformed into a float data type.

Normalize the data

Scaled data is typically required for models. The data is normalized from (0-255) to (0-1) in this code snippet, and the target variable is one-hot encoded for further analysis. The target variable has two kinds (0-1).

Augmentation

This leads us to the next stage Data augmentation is a type of data pre-processing. Frequently, the amount of data available is insufficient to execute the categorization assignment adequately. In these circumstances, we supplement the data. For instance, if we are dealing using a dataset categorizing scabies into two categories, We might not have a sufficient number of photos (since high-quality photos are difficult to get by). In this scenario, augmentation can be used to enhance the amount of your dataset. In image-based deep learning problems, augmentation is frequently employed to improve the volume and diversity of training data. Only the training set should be augmented, never the validation set. As you are aware, pooling enhances invariance. If a scabies image is in the image's upper left corner, you may use pooling to determine if the scabies is a little left/right/up/down in the upper left corner. However, given training data that includes data augmentation such as rotation, cropping, translation, lighting, scaling, noise addition, and so on, the model learns all of these changes. This considerably improves the model's accuracy. As a result, even if scabies appears in any part the model should be able to reproduce the image and identify it with great accuracy.

Convolutional neural network

Artificial intelligence has had significant results in the field of clinical diagnosis, while deep learning technology is critical in medical picture identification. A newcomer inside the field of artificial intelligence may struggle to determine which form of network to employ. There are several sorts of networks to select from, and new approaches are published and debated daily. To make matters worse, many neural networks are adaptable enough to perform (make a prediction) also when presented with incorrect data or a prediction difficulty. CNN approaches have increased classifier confidence, given quicker computation rates, and better outcomes. They were generally designed to identify just one or a few particular skins. A neural network using convolutions is just a deep learning approach that has been developed. gained prominence in many vision in computer applications and is gaining interest in a wide range of areas. A convolution neural network is composed of various building pieces, such as It employs convolution layers, pooling layers, and fully linked layers to learn spatial data hierarchies instantly and adaptively using a backpropagation approach. This remark from Dr. Prasad Samarakoon is one of the useful ways to understand convolutions: "A convolution may be thought of as "looking at a function's surrounds to make better/accurate forecasts of its output." Looking at tiny areas of a picture rather than the complete image at once to locate certain traits can be more successful. The CNN algorithm is composed of several modules which are organized in a certain workflow and are listed below:

- **Input Image**
- **Convolution Layer**
- **Pooling Layer**
- **Classification**

Input Image

CNN uses a picture as input, differentiates things with three color planes, and detects different color zones it also takes measurements of the picture. To demonstrate this procedure, we will use the RGB image shown below as an example.

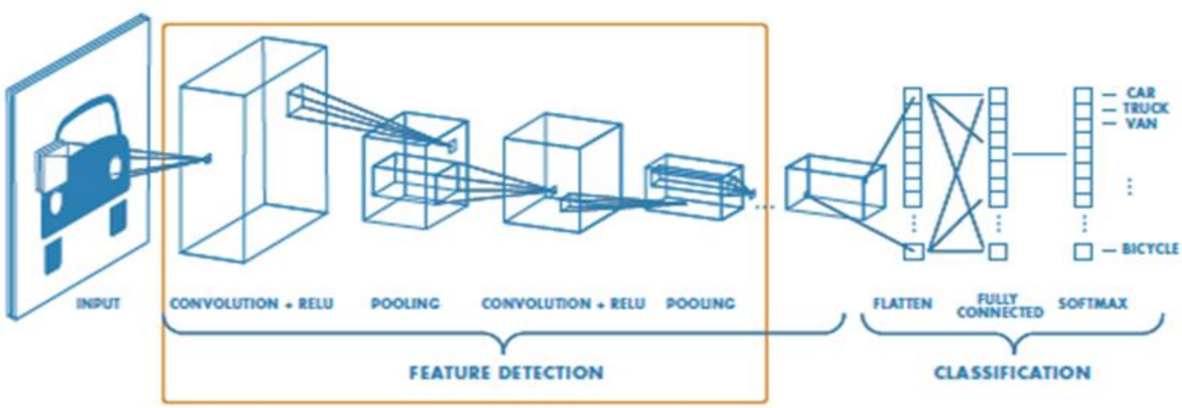

Figure 4 Convolutional Neural network structure/architecture

This picture contains a variety based on three main planes blue, green, and red, generally referred to as RGB. The several color palettes in which pictures can be found are then identified, such as RGB, CMYK, Grayscale, and many others It can be time-consuming to measure the picture dimensions, for, For instance, suppose the image is necessarily (*780x420*). One of CNN's useful qualities is that it decreases to make things easier, the image's dimensions to process while still retaining all of its elements in a single piece This has been completed to generate a more accurate prediction.

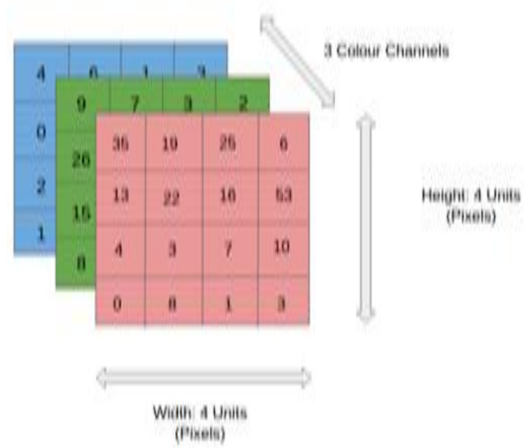

Figure 5 Layers of the image of RGB.

Convolution Layer

The layer of convolution is the major component according to CNN. It bears most of the computational burden on the network. This Cartesian coordinates the dot product of the two matrixes, one being the kernel set of trainable parameters while the other is the confined part of the perceptron. The kernel takes up less space than an image but is more detailed. This indicates that if a picture contains three (RGB) channels, the kernel length, and the breadth will be minimal, but the depths will span all three. The kernel performs a forward pass during a forward pass glides over the image's height and width, yielding input images of that responsive region This produces a two-dimensional image representation referred to as an activation mapping, which comprises the kernel's response at each picture spatial position. A step is the sliding size of the kernel. If we still have an intake of size W x W x D, as well as a doubt number, seeds per pod concerning spatial dimensions cushioning, stride S, and F amount P, we can compute calculating the output volume that uses the formula:

$$W_{out} = \frac{W - F + 2P}{S} + 1 \tag{3.1}$$

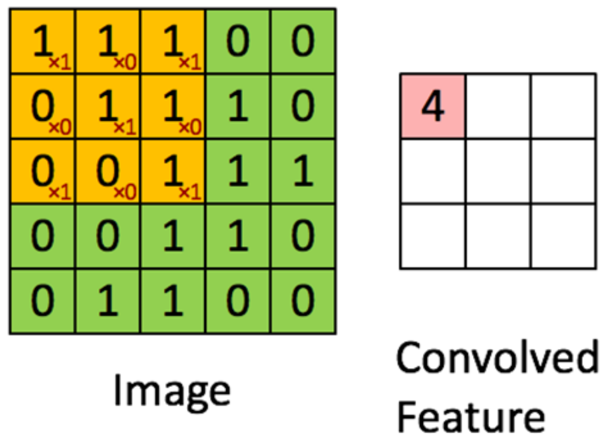

Figure 6 Convoluting a 5x5x1 image with a 3x3x1 kernel to get a 3x3x1 convolved feature

Pooling Layer

Pooling layers, also called downsampling, does dimensionality reduction to lower the set of variables in the input. Pooling, such as the layer of convolution, runs a filter applied to the full input but contains there are no weights. Rather, the kernel creates using an aggregate on the output array function to the receptive field data. There are two kinds of pooling:

Max pooling: As it advances the filter is applied to the pixel chosen by the input containing highest the value to send to the output vector. As a side note, this method is more regularly used employed than typical pooling.

Average pooling: The filter calculates the overall average in the receptive as it passes through the input field and sends it to the output array.

While a lot of data is wasted inside the pooling layer, it does provide several advantages for CNN. They aid in reducing complexity, increasing efficiency, and reducing the danger of overfitting.

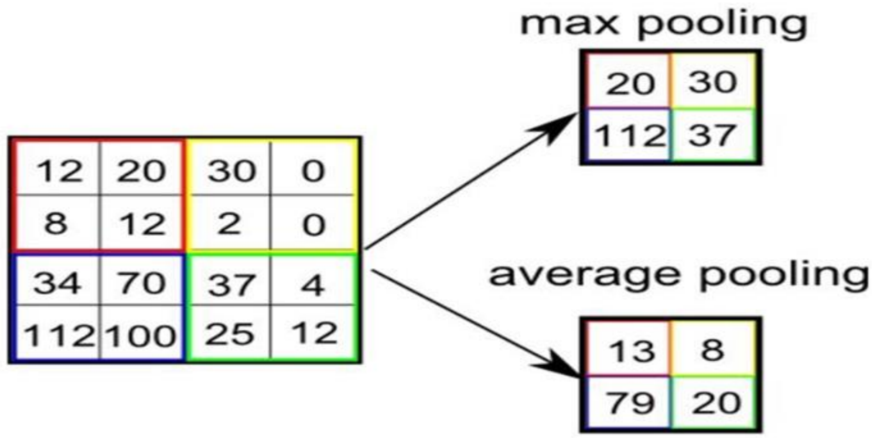

Figure 7 Pooling operations.

Fully-Connected Layer

The name of the full-connected layer accurately depicts its function. In layers that are only partially related, as previously stated, the input picture's pixel values are not directly related to the output layer. In the completely linked layer, however, each node in the output layer has a direct link to a component in the preceding layer. This layer categorizes the attributes received by the preceding levels and their numerous filters. Although convolution layers frequently employ ReLu functions for input classification, FC layers generally are using a softmax technique to provide likelihood between 0 and 1.

Activation Functions

In a neural network, an activation function defines how the network works weighted The aggregate of the inputs is converted into a result by a network layer of one or more nodes. An activation function is sometimes referred regarded as a "transfer function" at times. A narrow output range of the activation function is known as a "squashing function." Nonlinear activation functions are common, and this is known as "nonlinearity" in-network or layer design. The choice of The activation function does have a substantial impact influence on the neural network's capabilities and performance, and multiple It is possible to activate functionality. utilized in different regions of the model. Although systems are intended to use the same non-linear activation for any and each node in a layer, the activation function is used during or after processing information of every node on the network In a network, there are three sorts of layers: layers of input which accept original data from of the area concealed layers that accept input from some other layer, and then transmit output to that layer or output layers that provide a prediction. The same activation function is generally used by all buried levels. The output layer's activation function is frequently different from that of the hidden layers, and it is governed by the sort of forecast needed because of the model. Because activation functions are frequently mapping processes, the very initial derivative for an input data value may be calculated. This is required because neural networks were frequently trained to use the backpropagation on the error process, which necessitates updating the model's weights using the derivative of the prediction error.

There are several kinds of neural network activation functions, albeit only a few are likely to be employed in reality for the output and hidden layers.

Let's look at the activation functions for each type of layer one by one.

Why is an Activation Function Required in Neural Networks?

So now we understand what Activation Function is and what it accomplishes, but why do Neural Networks require it?

An activation function's goal is to add nonlinearity to the neural network.

Artificial neurons add an extra phase to forward propagation at each layer, but the computation is the same worthwhile. Here's why: Assume we have a neural network that doesn't have any activation functions.

In that situation, each neuron will just use the weights and biases to execute a linear change on the inputs. Because the composite of two linear functions is a linear function, it doesn't regardless of how many layers are added to a neural net; all levels will act the same way.

Regardless of how basic the neural network becomes, learning any tough task is difficult, and our model will be nothing more than multiple linear regression.

There are three prominent ones available to you might wish to explore for usage in hidden layers:

- **Rectified Linear Activation (ReLU)**
- **Logistic (Sigmoid)**
- **Hyperbolic Tangent (Tanh)**

This is not a full list of hidden layer activation functions, but it is the most prevalent.

Let's look more closely at each of them individually.

In inactivation that is nonlinear, the majority of functions are categorized depending on their ranges or their curvature.

Sigmoid Function

The curve of the Sigmoid Function is shaped like an S.

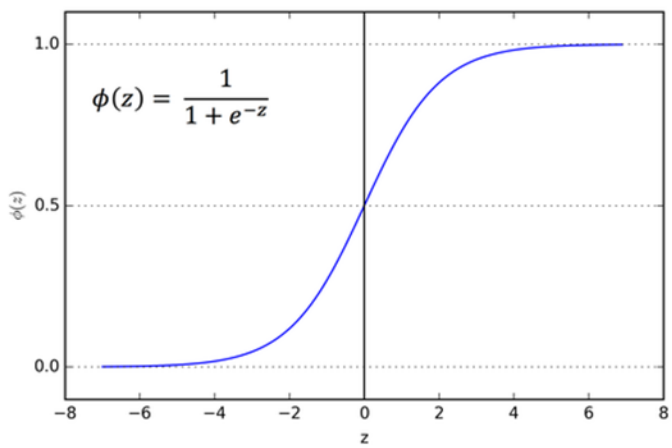

Figure 8 sigmoid activation functions.

A sigmoid function is applied mainly since it happens among (0 to 1). As just a result of this, It is especially beneficial for models required As an output, we can forecast the likelihood. Because everything possible occurs just between (0 to 1), the sigmoid is the best option.

The function can be differentiated.

That is, we can calculate the sigmoid curve's slope between any two locations.

The function itself is monotonic, but its derivative is not.

The logistic sigmoid functional can cause a neural network to malfunction network to become stuck during training.

The softmax feature is a broader function For classification tasks, logistic activation is utilized.

2. Activation Function for Tanh or Hyperbolic Tangent

Tanh is related to, but superior to, logistic sigmoid. The function does have a variety of between and (-1 through 1)Tanh is also sigmoidal (s-shaped).

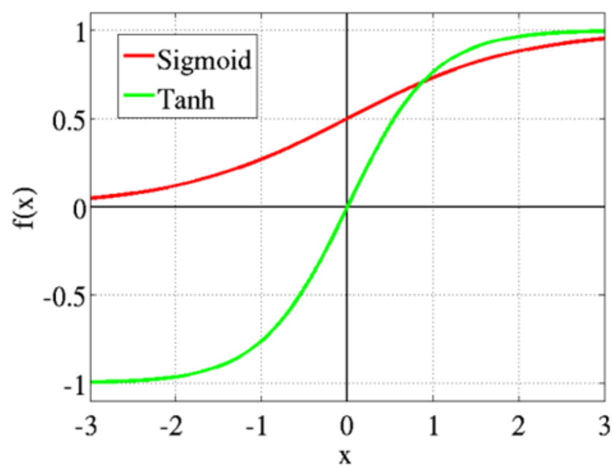

Figure 9 Hyperbolic Tangent (Tanh) Activation function.

The advantage is that negative inputs will appear on the graph. be substantially Negative or zero inputs will result in being near zero.

The function can be differentiated.

The function is monotonic, but its derivative is not.

The function is mostly used to differentiate between two classes.

In feed-forward networks, both logistic sigmoid activation functions are utilized.

3. ReLU (Rectified Linear Unit) Activation Function

Right present, The ReLU is by far the majority popular and widely employed world activation function. Since then, it has been put to use in Almost all deep learning and deep neural network approaches.

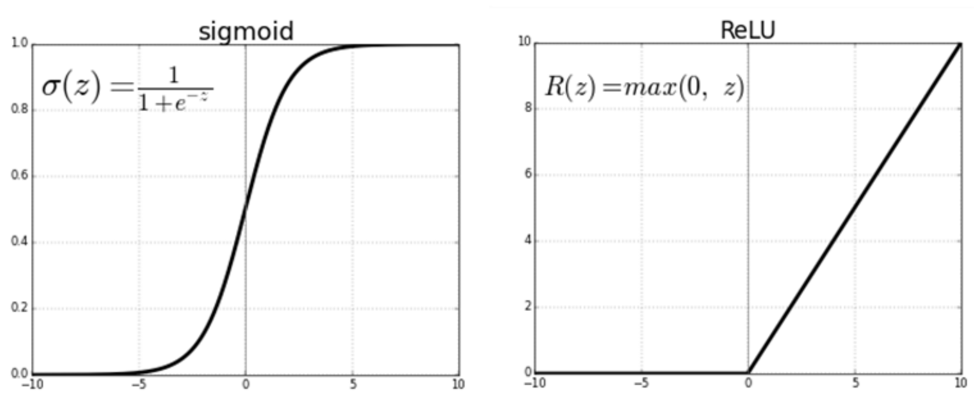

Figure 10 ReLU v/s Logistic Sigmoid.

As can be seen, the ReLU is only partially repaired (beginning at the bottom). When z is below zero, f (z) equals zero; when z is greater than or close to 0, f(z) equals z.

[From 0 to infinity]

Both the function and its derivative have a monotonic behavior.

However, any negative values are automatically converted to zero, reducing the model's ability to accurately fit or learn from data. That is, any input that is negative provided to a ReLU activation function instantly alters the value shown in the graph to zero, influencing the graph produced by just not properly Negative values mapped.

Loss Functions

The loss function is employed by machines that can learn. It is a means of assessing how successfully a particular algorithm imitates the provided data. If projections diverge too remote from reality outcomes, this loss function will provide a large number. Gradually, with the assistance Given a certain optimization function, The loss function develops to minimize prediction error. In this part, We will look at various loss functions & their applications for machine learning and deep learning. There is no such thing as a loss function that is one-size-fits-all for machine learning algorithms. There are several elements to consider when selecting a loss function for a given issue, including the machine learning technique algorithm used, the simplicity of computing derivatives, and, to some extent, the number of the data set's outliers. Depending on the type of learning activity loss functions are classified into two types. categories: categorization and regression losses. We strive to anticipate classification output from the collection of limited Given a big data collection containing category values, and pictures, classifying them into one of 0–1 digits. In contrast, regression is concerned with predicting a continuous value.

Regression Losses

Mean Square Error/Quadratic Loss/L2 Loss

$$MSE = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n} \tag{3.2}$$

Mean square error is defined as the square of the difference between forecasts as well as genuine observations. It is simply worried about the typical size of a mistake, regardless of its direction. However, because of squaring, forecasts that are far off from the real values are strongly punished in contrast to less deviating forecasts. MSE also has great mathematical aspects that make calculating gradients easier.

$$MAE = \frac{\sum_{i=1}^{n} \mid y_i - \hat{y}_i \mid}{n} \tag{3.3}$$

On either hand, mean absolute error is calculated as the total of the absolute discrepancies between predicted and observed values. This, like MSE, assesses the degree of error without taking into account its direction. MAE, unlike MSE, requires more complex techniques to compute gradients, such as linear programming. Furthermore, because it does not require a square, MAE is more resistant to outliers.

Mean Bias Error

This is far less prevalent in the machine learning sector than its equivalent. The main difference between this and MSE is that we avoid using absolute values. Both the positive and bad mistakes can cancel each other out, therefore extreme caution is advised. Despite being less precise In actuality, it may be utilized to determine whether or not the model possesses a bias, whether favorable may be negative. The formulation in mathematics:-

$$MBE = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)}{n} \tag{3.4}$$

Classification Losses

Hinge Loss/Multi-class SVM Loss

Simply expressed, the correct category's score ought to be greater by some margin than that of the total of both the erroneous individual category scores (usually one). As a result, Hinge loss is frequently Classification based on maximum margin, especially in support of support vector machines. But it is not distinguishable, this is a function defined, making it very simple to work with common Machine learning convex optimizers. The formulation in mathematics:

$$SVMLoss = \sum_{j \neq y_i} max(0, s_j - s_{y_i} + 1) \qquad (3.5)$$

Optimizers

Optimizers are techniques or approaches that are employed to decrease an error function (loss function) or to enhance production effectiveness Optimizers were mathematical functions that are used to solve problems. Affected by the learnable parameters of the model, such as Weights Optimizers help determine how to modify the neural network's weights adjusted learning rate to minimize losses

This post will go over optimizers and some popular ways.

Types of optimizers

Let's look at the many sorts of optimizers and how they work to minimize the loss function.

In this tutorial, we will learn about the following optimizers and consider the advantages and disadvantages of each method. As a result, after the course, It will indeed be able to contrast optimizers and the results. procedures upon which they are built.

Gradient Descent

Stochastic Gradient Descent

Stochastic Gradient descent with momentum

Mini-Batch Gradient Descent

Adagrad

RMSProp

AdaDelta

Adam

Before you continue, there are a few words you ought to be acquainted with.

Regularization

Practitioners of machine learning are often concerned about overfitting. Overfitting simply implies that your model predicts well on the data you used to train it but performs badly in the real world on fresh data. This can occur if one parameter is too weighted and ends up dominating the formula. Regularization is a phrase that has been introduced to the optimization process to assist avoid this.

Regularization involves the addition of a particular component to the loss function that penalizes big weight values. That is, in addition to being punished for inaccurate predictions, you will be penalized for having high weight values even if your forecasts are correct. This just ensures that your weights remain on the lower side and, as a result, generalize better to new data.

Stochastic Gradient Descent

Rather than computing the gradients for all of your training instances on each gradient descent run, it is sometimes more effective to merely utilize a subset of the training examples each time. Stochastic Gradient Descent is a method that employs either batch of samples at a time or random examples on each run.

Because we're attempting to explain things intuitively, we haven't provided the formal functions for the ideas in this piece.

Other varieties of optimizers

It's impossible to emphasize how widespread gradient descent is because it's utilized everywhere from simple spreadsheets to intricate neural net structures (backpropagation is Gradient Descent implemented on a network). However, there are various sorts of Gradient Descent optimizers that are employed, and here are a few of them:

Adagrad

Because Adagrad tailors the learning rate to individual features, some of the weights in your dataset will have different learning rates than others. This works particularly effectively for sparse datasets with a large number of missing input samples. However, Adagrad has a significant issue: the adaptive learning rate tends to decrease with time. Some of the optimizers listed below are attempting to solve this issue.

RMSprop

RMSprop is a variant of Adagrad created by Professor Geoffrey Hinton for his neural networks class. Instead of allowing all gradients to accumulate for momentum, it simply allows gradients in a specified window to accrue. RMSprop is similar to Adaprop, another optimizer that attempts to address some of the difficulties that Adagrad left unresolved.

Adam

Adam represents the adaptive moment estimation Which is another way of computing current gradients from earlier gradients. Adam also uses momentum by combining fractions of previous gradients with the current one. This optimizer has gained popularity and is now widely used in neural network training.

It's easy to become overwhelmed by the sophistication of some of these new optimizers. Just keep in mind that they all have the same goal: to reduce our loss function. Even the most complicated methods are fundamentally basic.

Approach to Transfer Learning

Our goal is to train a convolutional neural network (CNN) to recognize objects in photos. As a result, rather than creating and training the CNN from the ground up, we'll employ a model that has already been constructed and trained with transfer learning.

Transfer learning's core assumption is uncomplicated: Apply a model that's been trained on the huge dataset. its knowledge to a more limited dataset We freeze our network's initial convolutional layers for object recognition and only train the last several levels that produce a prediction. Convolutional layers extract basic, low-level characteristics that are relevant across pictures — such as edges, patterns, and gradients — while subsequent layers detect particular aspects within an image, such as eyes or wheels. As a result, we may use a network trained on unrelated categories in a big dataset (often Imagenet) for our problem since photos share universal, low-level properties. The photos in the dataset are quite similar to those in the Imagenet dataset, thus any expertise gained on Imagenet should readily transfer to this challenge.

Fine Tuning Off-the-shelf Pre-trained Models

This is a much more interesting strategy in which we not only rely on the characteristics retrieved derived from pre-trained algorithms to update the last layer while selectively retraining others of the earlier layers.

Networks of neurons are multi-layered architectures that have several configurable hyperparameters. The earliest layers capture generic information, while the subsequent layers concentrate on the particular job at hand. To make the representations of higher-order features in the basic more suitable model for the current job, it becomes appropriate to fine-tune them. We can retrain parts of the model's layers while leaving others locked in training.

The following picture depicts an object identification job in which the bottom network layers learn relatively general characteristics and the higher levels acquire extremely task-specific attributes.

Fine-tuning: Domain adaption under supervision

Freezing vs. Fine-tuning

Re-training (or "fine-tuning") With the training of a classifier you added, the weights of the upper pre-trained model layers are one obvious strategy to improve the model's performance even more. This forces the weights are being updated using generic feature mappings learned by the model derived from the originator job. Fine-tuning would be required to enable the prototype to use previous knowledge inside the target area and some re-learning.

Furthermore, rather than fine-tuning the entire model, one should strive to fine-tune a limited number of upper layers Those few layers discover basic and generic characteristics that may be applied to nearly any form of data.

As a result, it is prudent to freeze & reuse those layers of the fundamental information gained from previous training. As we go in the hierarchy, the characteristics become more particular to that same dataset used to train a model Instead of overwriting the general learning, fine-tuning seeks to customize these specialized characteristics to operate because of the new dataset.

Transfer Learning for Deep Learning: Freeze vs. Fine-Tune

Transferring knowledge in six phases

Finally, let us guide you through In practice, how does transfer learning work?

Process of learning transfer

1. Get such a pre-trained model.

The initial stride is to decide which pre-programmed type we want to use as the foundation depends upon the nature of our training assignment. To be compatible Transfer learning necessitates a significant link between the pre-trained source model's knowledge and the destination domain of the task.

Here are some examples of pre-trained models:

In terms of computer vision:

- VGG 16
- VGG 19
- Inception- V3
- X.Ception
- ResNet 50

For NLP tasks:

- Word.2Vec
- Glo.Ve
- Fast.Text

The reason that I am using VGG16 is the object identification and classification method that can accurately identify 1000 photos from 1000 distinct categories. It is a common picture classification technique that works well with transfer learning and we have a simple dataset in which we do not have many details.

2. Create a base model

The foundation design is among the structures, for example, VGG-16, that we choose in the initial phase since it is closely related to our purpose. We have the option of downloading the network weights, saving time on extra model training.

The base model may contain we need additional neurons inside the output data layer for our application. In such cases, we must eliminate the output data layer and make the necessary changes.

3. Freeze layers

To prevent the additional work of having the model learn the fundamental characteristics, it is necessary to freeze the initial layers from the pre-trained model.

When we cannot freeze the earliest layers, We shall be defeated. everything of the previous learning. This is equivalent to preparing the model from start and represents a waste of time, money, and other resources.

4. Create new trainable layers

The feature extraction layers are the only knowledge we reuse from the basic model. To forecast the model's specific duties, we must put more layers on top of them. These are the last output layers in most cases.

5. Train the new layers

The ultimate output of the pre-trained model will almost certainly differ from the result we desire for our model. Models that have been pre-trained using the Dataset from Image Net, for example, will produce 1000 different categories.

However, the model must apply to two groups. In this instance, we must train that the model has been updated with the new output layer.

6. Fine-tune your model

Fine-tuning is one approach to enhancing performances.

Fine-tuning entails unfreezing a portion and training the whole model on the basic model complete dataset again with a relatively slow pace of learning. The slow learning rate will boost the model's performance just on the dataset while limiting the fitting problem.

CHAPTER IV

Implementation and Results

Implementation

The fundamental goal of this research is to create a sophisticated expert system that uses deep CNN to classify skin lesions. Python 2018 is used to implement the suggested intelligent expert system. Experiments are carried out on an Intel(R) Intel(R) Core(TM) i3-4005U CPU running Microsoft Windows 10, 64 bits at 1.70GHz. In the ratio of 70:20:10, the dataset is separated into three sections: training data, testing data, and validation data. As a result, Convolutional Neural Networks may be implemented on any device. However, Python (Python 3.5 in this case) is favored since It gives the developer access to a diverse set of machine learning and neural network packages. The following major libraries were utilized in implementation:

a) OpenCV: OpenCV is a Python-based Open Cv library that also has Java, C, and C++ interfaces. It works on a variety of systems which is employed in real-time computer vision and image processing.

b) Scikit learn is A free computer vision package that includes algorithms for regression & classification.

c) Keras: This is a supervised learning package that can operate in addition to TensorFlow.

d) Tensorflow: Software is an open library developed by Google. It is utilized as a Keras backend because it is beneficial for numerical calculations and calculations. NumPy, pandas, and other libraries are also utilized.

Dataset

Dataset We used the Tripoli Hospital University, a vast collection of multi-source dermatoscopic pictures of common scabies skin lesions obtained and archived by diverse modalities and from a distinct population, in this work. The collection comprises 138 photos of two forms of scabies: normal and abnormal. The figure depicts sample photos of all scabies varieties.

Figure 11 Scabies/Abnormal /Normal Dermoscopy images.

To authenticate our model, we separated our dataset into training phase 70% and validation data 10%. The photos' original dimension was 600x450 pixels, which were downsampled to 224x224 pixel resolution to make them suitable for our training model. To accommodate a high number of photos in the collection, the photos have been rotated in every direction (each by 45 degrees) and then flipped.

Experimental of the First Model

The picture is then sent into the network's first layer as an input. The Convolutional Neural Network is then to it, as described above, till high-level qualities such as edge, border, and color are generated. This is performed by employing several ConvNet methods including such Pooling layer, Convolution Layers, and others until the image flattens out from an image vector. Those would be since the vectors which can be utilized for categorization contain information that can be used to determine high-level characteristics. We categorize skin lesion test photos and calculate accuracy after training our dataset with CNN and Transfer learning with the VGG16 model. We trained our models for a total of 20 epochs. The model's output is computed to use the same training and validation photos. In our study, 70 photographs are utilized the remaining 43 photos are utilized for testing and training. As seen in Figure 4, the transferred learning VGG16 model is utilized as a feature extractor to improve performance. Transfer learning using VGG16 with reduced loss and higher accuracy was explored, and accuracy is enhanced for the transfer learning model linked to the CNN model. We observed that our dataset's photos are divided into two groups.

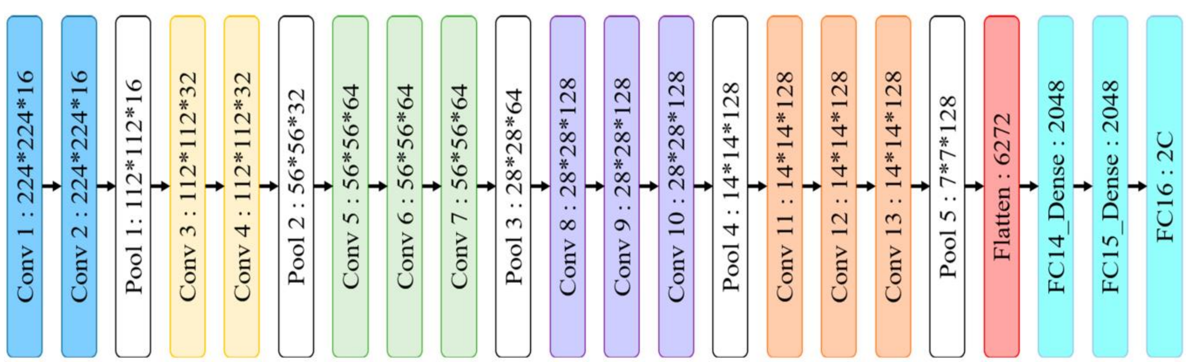

Figure 12 depicts the VGG16 model setup that utilizes the first dataset to learn features and identify the presence of scabies photos.

According to Figs above, the conclusion is that freezing all previous layers and solely training the top layer results in greater accuracy. By freezing all previous convolutional layers, VGG16 is learned. Only the extra top layer weights are trained and tweaked.

Results of the First Model

Loss and accuracy curves to evaluate the effectiveness of the planned model, we created a loss and accuracy curve. These charts indicate that as the number of iterations increases, accuracy increases and loss decreases. The difference between the validation and training curves indicates that the model is well fitted, and it will perform well on unknown pictures. Figure 4.2 depicts the accuracy curve, whereas Figure 4.3 depicts the loss curve.

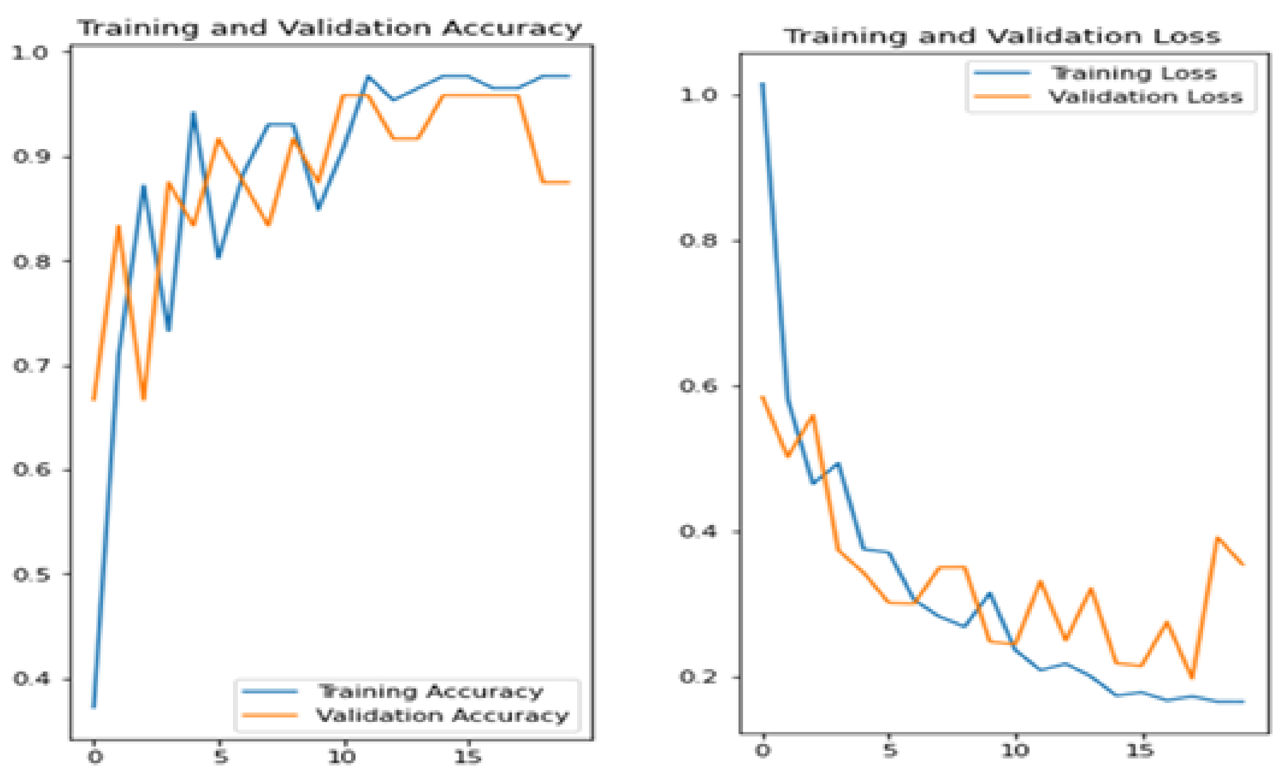

Figure 13 Accuracy and Loos Per Epoch of the First Model.

as shown in Figure 15 the assessment parameters of the analysis that was trained using the dataset to the six photos are shown below, along with the percentage of accuracy and predictions of the images for each image a. The model has 97.67 percent accuracy.

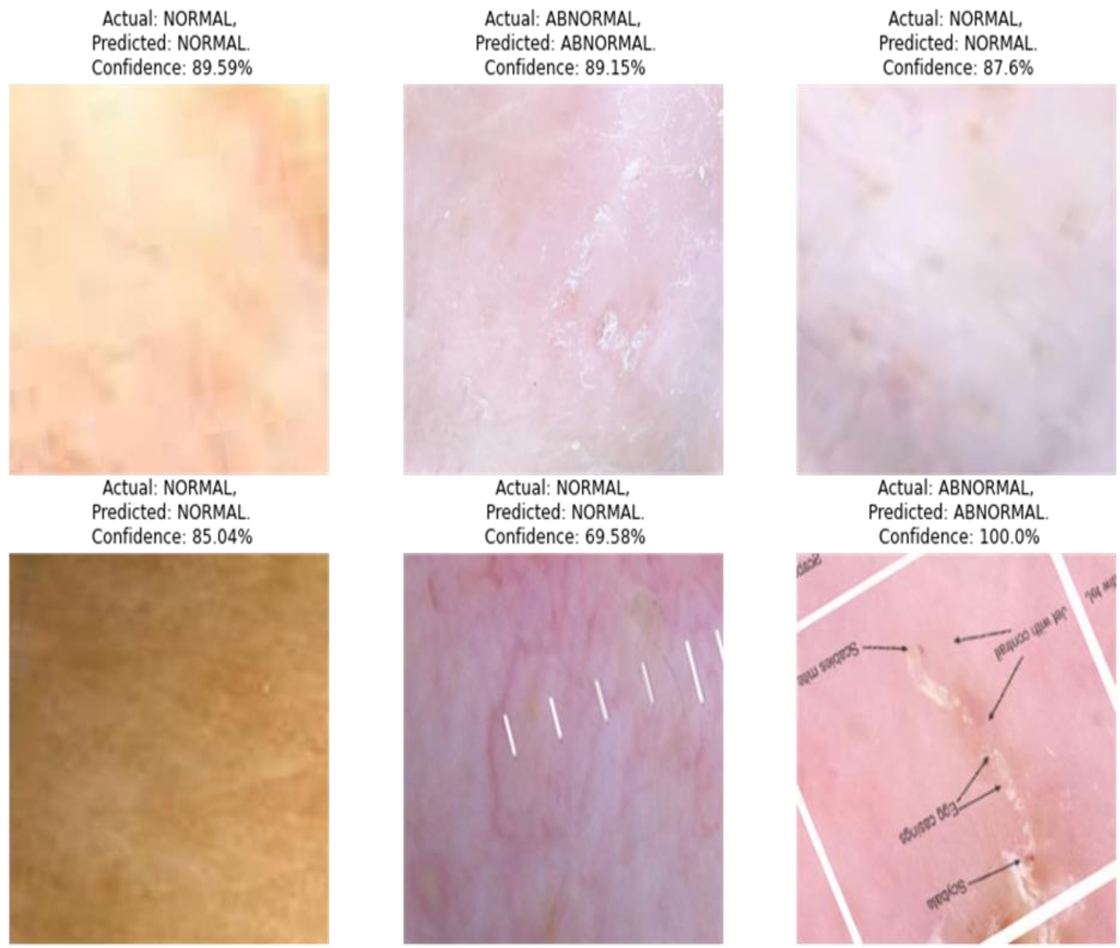

Figure14: above depicts the evaluation results of the First Model.

The quality of this model can be enhanced by adding more training data, but for now, we have everything we need to enable us to develop the model that categorized Normal and Abnormal instances from Dermoscopy images.

Experimental of the Second Model

The picture is then sent into the network's first layer as an input. The Convolutional Neural Network is then to it, as described below, till high-level qualities such as edge, border, and color are generated. This is performed by employing several ConvNet methods including such Pooling layer, Convolution Layers, and others until the image flattens out from an image vector. Those would be since the vectors which can be utilized for categorization contain information that can be used to determine high-level characteristics. We categorize skin lesion test photos and calculate accuracy after training our dataset with CNN and the Traditional model. We trained our models for a total of 20 epochs. The model's output is computed to use the same training and validation photos. In our study, 70% of photographs are utilized the remaining 30% of photos are utilized for testing and training.

conv2d_48 (Conv2D)	(None, 254, 254, 32)	896
max_pooling2d_48 (MaxPooling2D)	(None, 127, 127, 32)	0
conv2d_49 (Conv2D)	(None, 125, 125, 64)	18496
max_pooling2d_49 (MaxPooling2D)	(None, 62, 62, 64)	0
conv2d_50 (Conv2D)	(None, 60, 60, 128)	73856
max_pooling2d_50 (MaxPooling2D)	(None, 30, 30, 128)	0
conv2d_51 (Conv2D)	(None, 28, 28, 256)	295168
max_pooling2d_51 (MaxPooling2D)	(None, 14, 14, 256)	0
conv2d_52 (Conv2D)	(None, 12, 12, 412)	949660
max_pooling2d_52 (MaxPooling2D)	(None, 6, 6, 412)	0
conv2d_53 (Conv2D)	(None, 4, 4, 582)	2158638
max_pooling2d_53 (MaxPooling2D)	(None, 2, 2, 582)	0
flatten_16 (Flatten)	(None, 2328)	0
dense_32 (Dense)	(None, 64)	149056
dense_33 (Dense)	(None, 2)	130

Figure 15 depicts the traditional model setup that utilized the dataset to learn features and identify the presence of scabies photos.

Results of the Second Model

Loss and accuracy curves to evaluate the effectiveness of the planned model, we created a loss and accuracy curve. These charts indicate that as the number of iterations increases, accuracy increases and loss decreases. The difference between the validation and training curves indicates that the model is well fitted, and it will perform well on unknown pictures. Figure 4.2 depicts the accuracy curve, whereas Figure 4.3 depicts the loss curve.

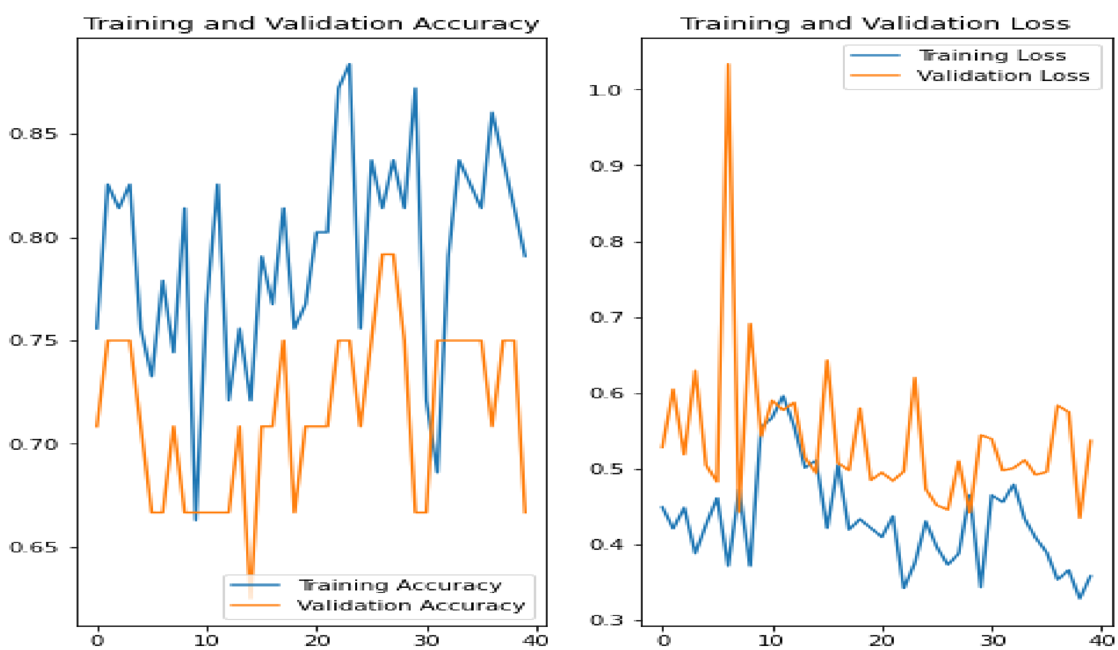

Figure 16 Accuracy and Loos Per Epoch of the Second Model.

as shown in Figure 16 the assessment parameters of the analysis that was trained using the dataset to the six photos are shown below, along with the percentage of accuracy and predictions of the images for each image a. The model has 97.67 percent accuracy.

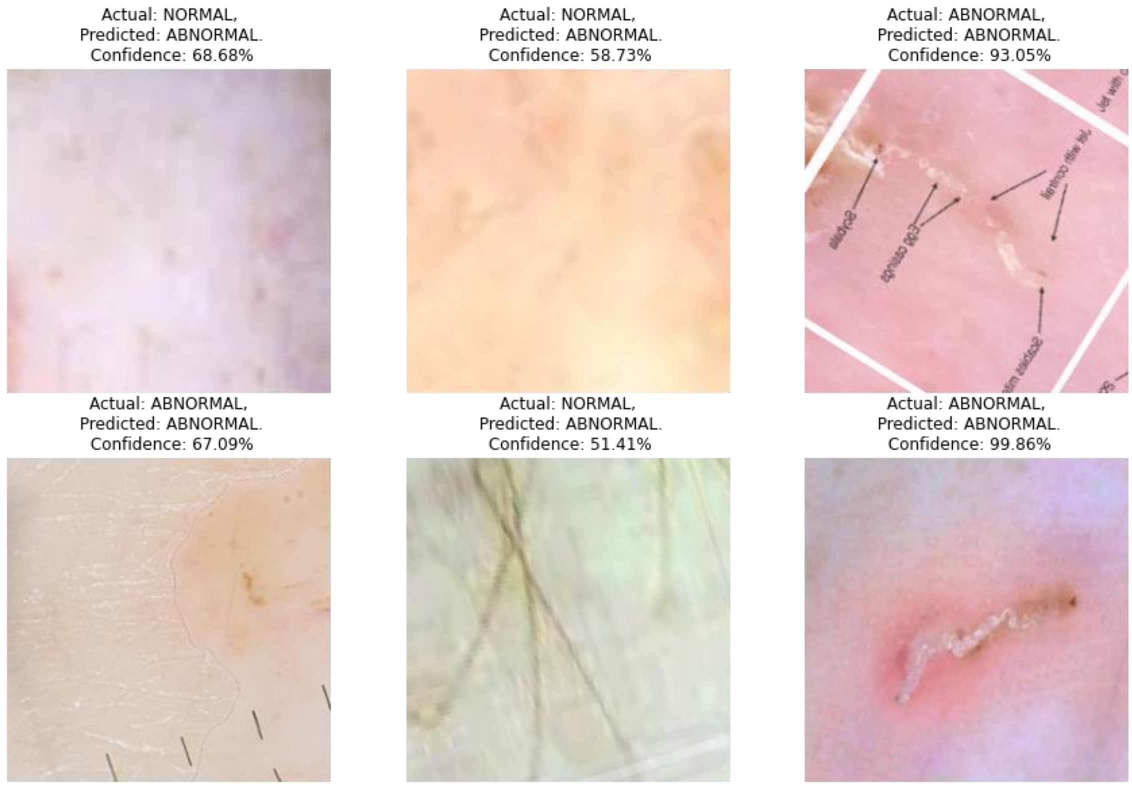

Figure17: above depicts the evaluation results of the Second Model.

The quality of this model can be enhanced by adding more training data, but for now, we have everything we need to enable us to develop the model that categorized Normal and Abnormal instances from Dermoscopy images.

Comparison Between Two Models

Because of the balanced data distribution, it was simple to train the model properly, and the rate of loss was reduced. In such instances, the proposed method is a highly beneficial strategy for balancing data. To equalize the data, we employed enhanced augmentation for both the minority classes. Following data balancing, we import and fine-tune the learning model to train on our dataset. as we can see below Table2We divided the dataset into 70/20/10 train/test/validation splits and utilized 20 epochs. The Adam optimizer, category cross-entropy, and accuracy were used to assess the Model's performance. Model performance was validated by calculating training and validation performance, the loss and accuracy curve, and comparing it to current research. A. Validation of the Model The model was validated using 10% photos from the validation dataset. We estimated the weighted average accuracy to better understand the model's generalized performance. The weighted average accuracy for the two scabies classes was estimated to be 0.9767 percent. The next section contains a full two-class classification report. We also compared our model's performance to that of current research.

Table 1 Compassion of two Model learning parameters.

Models	**VGG-16**	**Traditional**
Parameters	**Value**	**Value**
Epoch	20	40
Learning Rate	0.0001	0.0001
Training Time	9 min	14 min
Training Accuracy	97.67 %	88.37 %
Testing Accuracy	95.34 %	84.02%

VGG16 outperformed the traditional model in terms of accuracy. Training and testing duration, as well as the processing capability of the simulating machine, all had a role in training and testing accuracy. Furthermore, the VGG-16 and traditional mode accuracy reached 97.67% and 88.37%, respectively.

The information we handle can be systematized and organized in many ways. One is through comparative tables. They are general schemes with which it is possible to contrast the different elements that make up a certain theme with another similar one. This type of scheme fulfills a triple function: 1) provides a synthesis of information, 2) compares the relevant data, and 3) the similarities and differences of a theme as shown in **Table 2** below.

Table2: compares the suggested technique to several previously published publications.

AUTHORS	DISEASE	SOFTWARE	ACCURACY	NUMBER OF DATA	AIM
Isaac Kofi Nit El al "2019"	atopic dermatitis, acne vulgaris,& scabies	CNN	88% , 85%, and 84.7%	102, 87, and65.	Data collection &analysis of Three common skin diseases.
Bacon, F. S El al “2009”	Scabies, Acne, & Vulgar.	ANN	90%	-	Show how CNN works, data collection, and analysis.
Proposed	**Scabies disease**	**CNN**	**97.67%**	**138**	**An approach that effectively diagnoses scabies (NORMAL or ABNORMAL).**

Based on **Table2** such comparison, we may infer that our suggested strategy outperforms the current studies.

CHAPTER V

Conclusion and Recommendations

Conclusion

Deep learning has proven to be quite successful at detecting scabies. Various networks were employed in this study's testing and training. VGG-16 was used to test and train two distinct classifications. This study's accuracy, specificity, and sensitivity results show that VGG-16 (97.67 percent). This study included a comparison of our study to other studies to confirm the study's validity. However, numerous factors can influence the effectiveness of the output, including the processing speed of the simulating machine and other interferences that may occur. Deep CNN learning refers to a family of AI approaches that use more dynamic representations of data to carry out a certain job. These methods make use of planning data to determine how these depictions should be made in a way that is appropriate for the particular task. Traditional AI, on the other hand, uses meticulously constructed highlights to create representations of the information that is used to carry out the task. In a variety of situations, deep learning has been shown to outperform conventional machine learning algorithms and is being used to modify existing therapeutic practices. Convolutional neural networks have outperformed expectations in many image interpretation tasks and may be able to extract more info from histopathology images.

Recommendations

Dermatology diagnosis requires a high level of accuracy, and the overall performance of a CNN network must be extremely well tuned. However, the results of this investigation revealed that VGG-16 has a very high-performance rate. It is necessary to investigate greater optimization outcomes and performance. Several combination strategies can be employed to improve the findings of this investigation. As a combination approach, Support Vector Machine (SVM) may produce extremely successful results. Previous research has shown that ResNet50-SVM and VGG19-SVM may achieve very high performance during testing and training.

References

Ahmad, B., Usama, M., Huang, C. M., Hwang, K., Hossain, M. S., & Muhammad, G. (2020). Discriminative feature learning for skin disease classification using a deep convolutional neural network. *IEEE Access*, *8*, 39025-39033.

Akyeramfo-Sam, S., Philip, A. A., Yeboah, D., Nartey, N. C., & Nti, I. K. (2019). A web-based skin disease diagnosis using convolutional neural networks. International Journal of Information Technology and Computer Science, 11(11), 54-60.

Aloysius, N., & Geetha, M. (2017, April). A review on deep convolutional neural networks. In 2017 international conference on communication and signal processing (ICCSP) (pp. 0588-0592). IEEE.

Alzahrani, S., Al-Nuaimy, W., & Al-Bander, B. (2019, October). Seven-point checklist with convolutional neural networks for melanoma diagnosis. In *2019 8th European Workshop on Visual Information Processing (EUVIP)* (pp. 211-216). IEEE.

Arik, A., Gölcük, M., & Karslıgil, E. M. (2017, May). Deep learning-based skin cancer diagnosis. In *2017 25th Signal Processing and Communications Applications Conference (SIU)* (pp. 1-4). IEEE.

Back, S., Lee, S., Shin, S., Yu, Y., Yuk, T., Jong, S., ... & Lee, K. (2021). Robust Skin Disease Classification by Distilling Deep Neural Network Ensemble for the Mobile Diagnosis of Herpes Zoster. *IEEE Access*, *9*, 20156-20169.

Bajwa, M. N., Muta, K., Malik, M. I., Siddiqui, S. A., Braun, S. A., Homey, B., ... & Ahmed, S. (2020). Computer-aided diagnosis of skin diseases using deep neural networks. Applied Sciences, 10(7), 2488.

Baweja, H. S., & Parhar, T. (2016, July). Leprosy lesion recognition using convolutional neural networks. In *2016 International Conference on Machine Learning and Cybernetics (ICMLC)* (Vol. 1, pp. 141-145). IEEE.

Brinker, T. J., Hekler, A., Utikal, J. S., Grabe, N., Schadendorf, D., Klode, J., ... & Von Kalle, C. (2018). Skin cancer classification using convolutional neural networks: a systematic review. Journal of medical Internet research, 20(10), e11936.

Brownlee, J. (2019). Deep learning for computer vision: image classification, object detection, and face recognition in python. Machine Learning Mastery.

Chen, Y., Zhu, K., Zhu, L., He, X., Ghamisi, P., & Benediktsson, J. A. (2019). Automatic design of convolutional neural network for hyperspectral image classification. IEEE Transactions on Geoscience and Remote Sensing, 57(9), 7048-7066

Dabiri, S., & Heaslip, K. (2018). Inferring transportation modes from GPS trajectories using a convolutional neural network. Transportation research part C: emerging technologies, 86, 360-371.

De, A., Sarda, A., Gupta, S., & Das, S. (2020). Use of artificial intelligence in dermatology. *Indian Journal of Dermatology*, *65*(5), 352.

Dos Santos, F. L. C., Paci, M., Nanni, L., Brahnam, S., & Hyttinen, J. (2015). Computer vision for virus image classification. Biosystems Engineering, 138, 11-22.

Fidan, U., Sarı, İ., & Kumrular, R. K. (2016, October). Classification of skin lesions using ANN. In the *2016 Medical Technologies National Congress (TIPTEKNO)* (pp. 1-4). IEEE.

Göçeri, E. (2020, December). Convolutional neural network-based desktop applications to classify dermatological diseases. In 2020 IEEE 4th International Conference on Image Processing, Applications and Systems (IPAS) (pp. 138-143). IEEE.

Guha, S. R., & Haque, S. R. (2019, December). Convolutional neural network-based skin lesion analysis for classifying melanoma. In 2019 International Conference on Sustainable Technologies for Industry 4.0 (STI) (pp. 1-5). IEEE.

Hameed, N., Shabut, A. M., & Hossain, M. A. (2018, December). Multi-class skin diseases classification using a deep convolutional neural network and support vector machine. In *2018 12th International Conference on Software, Knowledge, Information Management & Applications (SKIMA)* (pp. 1-7). IEEE.

Han, S. S., Moon, I. J., Lim, W., Suh, I. S., Lee, S. Y., Na, J. I., ... & Chang, S. E. (2020). Keratinocyte skin cancer detection on the face using region-based convolutional neural network. JAMA Dermatology, 156(1), 29-37.

He, X., Wang, S., Shi, S., Tang, Z., Wang, Y., Zhao, Z., ... & Chu, X. (2019, December). Computer-Aided Clinical Skin Disease Diagnosis Using CNN and Object Detection Models. In *2019 IEEE International Conference on Big Data (Big Data)* (pp. 4839-4844). IEEE.

Hijazi, S., Kumar, R., & Rowen, C. (2015). Using convolutional neural networks for image recognition. Cadence Design Systems Inc.: San Jose, CA, USA, 9.

Iranpoor, R., Mahboob, A. S., Shahbandegan, S., & Baniasadi, N. (2020, December). Skin lesion segmentation using convolutional neural networks with improved U-Net architecture. In *2020 6th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS)* (pp. 1-5). IEEE.

Karn, U. (2016). An intuitive explanation of convolutional neural networks. *The data science blog*.

Khan, Z. A., Hussain, T., Ullah, A., Rho, S., Lee, M., & Baik, S. W. (2020). Towards efficient electricity forecasting in residential and commercial buildings: A novel hybrid CNN with an LSTM-AE based framework. Sensors, 20(5), 1399.

Kim, D. H., & MacKinnon, T. (2018). Artificial intelligence in fracture detection: transfer learning from deep convolutional neural networks. Clinical radiology, 73(5), 439-445.

Kumar, V. B., Kumar, S. S., & Saboo, V. (2016, September). Dermatological disease detection using image processing and machine learning. In *2016 Third International Conference on Artificial Intelligence and Pattern Recognition (AIPR)* (pp. 1-6). IEEE.

Liu, N., Bai, Y., Li, X., & Zhang, Y. (2022). Scabies Knowledge Among Undergraduate Nursing Students in China: A Questionnaire Survey. Clinical, Cosmetic and Investigational Dermatology, 15, 133.

Madhurshalini, M., Nair, C., & Goel, N. (2020, September). Automatic identification of skin lesions using deep learning techniques. In *2020 IEEE/ITU International Conference on Artificial Intelligence for Good (AI4G)* (pp. 230-235). IEEE.

Namozov, A., & Im Cho, Y. (2018, October). Convolutional neural network algorithm with parameterized activation function for melanoma classification. In *2018 International Conference on Information and Communication Technology Convergence (ICTC)* (pp. 417-419). IEEE.

Namozov, A., Ergashev, D., & Im Cho, Y. (2018, December). Adaptive activation functions for skin lesion classification using deep neural networks. In 2018 Joint 10th International Conference on Soft Computing and Intelligent Systems (SCIS) and 19th International Symposium on Advanced Intelligent Systems (ISIS) (pp. 232-235). IEEE.

Postelnicu, R. (2020). A Comparison between E-learning Resources currently used in Project Management Online Training. ENTRENOVA-ENTerprise REsearch InNOVAtion, 6(1), 410-421.

QI, X. (2017). The understanding of convolutional neuron network family. DEStech Transactions on Computer Science and Engineering,(ii).

Rathod, J., Waghmode, V., Sodha, A., & Bhavathankar, P. (2018, March). Diagnosis of skin diseases using Convolutional Neural Networks. In *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)* (pp. 1048-1051). IEEE.

Reis, P. M. L. D. S. (2022). Data Labeling tools for Computer Vision: a Review. Rimi, T. A., Sultana, N., & Foysal, M. F. A. (2020, May). Derm-in: Skin diseases detection using convolutional neural network. In *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)* (pp. 1205-1209). IEEE.

Rimi, T. A., Sultana, N., & Foysal, M. F. A. (2020, May). Derm-NN: skin diseases detection using convolutional neural network. In 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS) (pp. 1205-1209). IEEE.

Rodrigues, J. F., Brandoli, B., & Amer-Yahia, S. (2020, July). DermaDL: advanced Convolutional Neural Networks for automated melanoma detection. In 2020 IEEE 33rd International Symposium on Computer-Based Medical Systems (CBMS) (pp. 504-509). IEEE.

Singh, V., & Nwogu, I. (2018, October). Analyzing skin lesions in dermoscopy images using convolutional neural networks. In 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (pp. 4035-4040). IEEE.

Usama, M., Ahmad, B., Xiao, W., Hossain, M. S., & Muhammad, G. (2020). Self-attention-based recurrent convolutional neural network for disease prediction using healthcare data. Computer methods and programs in biomedicine, 190, 105191.

Wang, F., Yang, J. F., Wang, M. Y., Jia, C. Y., Shi, X. X., Hao, G. F., & Yang, G. F. (2020). Graph attention convolutional neural network model for chemical poisoning of honey bees' prediction. Science Bulletin, 65(14), 1184-1191.

Wiley, V., & Lucas, T. (2018). Computer vision and image processing: a paper review. International Journal of Artificial Intelligence Research, 2(1), 29-36.

Wu, Z., Zhao, S., Peng, Y., He, X., Zhao, X., Huang, K., ... & Li, Y. (2019). Studies on different CNN algorithms for face skin disease classification based on clinical images. *IEEE Access*, *7*, 66505-66511.

Yang, S. J., Lipnick, S. L., Makhortova, N. R., Venugopalan, S., Fan, M., Armstrong, Z., ... & Rubin, L. L. (2019). Applying deep neural network analysis to high-content image-based assays. SLAS DISCOVERY: Advancing Life Sciences R&D, 24(8), 829-841.

Zhang, B., Wang, Z., Gao, J., Rutjes, C., Nufer, K., Tao, D., ... & Menzies, S. W. (2020). Short-term lesion change detection for melanoma screening with a novel siamese neural network. IEEE transactions on medical imaging, 40(3), 840-851.

Zuo, H., Fan, H., Blasch, E., & Ling, H. (2017). Combining convolutional and recurrent neural networks for human skin detection. IEEE Signal Processing Letters, 24(3), 289-293.

O'Shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. arXiv preprint arXiv:1511.08458.

Lee, H., & Song, J. (2019). Introduction to a convolutional neural network using Keras; an understanding from a statistician. Communications for Statistical Applications and Methods, 26(6), 591-610.

Jacovi, A., Shalom, O. S., & Goldberg, Y. (2018). Understanding convolutional neural networks for text classification. arXiv preprint arXiv:1809.08037.

Kuo, C. C. J. (2016). Understanding convolutional neural networks with a mathematical model. Journal of Visual Communication and Image Representation, 41, 406-413.

Spencer Jr, B. F., Hoskere, V., & Narazaki, Y. (2019). Advances in computer vision-based civil infrastructure inspection and monitoring. Engineering, 5(2), 199-222.

Buch, N., Velastin, S. A., & Orwell, J. (2011). A review of computer vision techniques for the analysis of urban traffic. IEEE Transactions on intelligent transportation systems, 12(3), 920-939.

Mopuri, K. R., Garg, U., & Babu, R. V. (2018). Cnn fixations: an unraveling approach to visualize the discriminative image regions. IEEE Transactions on Image Processing, 28(5), 2116-2125.

Kuo, C. C. J. (2016). Understanding convolutional neural networks with a mathematical model. Journal of Visual Communication and Image Representation, 41, 406-413.

Saha, S. (2018). A comprehensive guide to convolutional neural networks—the ELI5 way. Towards data science, 15.

Jiang, Z. P., Liu, Y. Y., Shao, Z. E., & Huang, K. W. (2021). An Improved VGG16 Model for Pneumonia Image Classification. *Applied Sciences*, *11*(23), 11185.

APPENDICES

Appendix A

MODEL CODE

```
import TensorFlow as tf

from TensorFlow.Keras import models, layers

import matplotlib.pyplot as plt

import split folders

input_folder = 'C:/Users/SPACE/Desktop/Scabies-dataset-turns

split folders.ratio(input_folder, output="datasetak71",

                seed=42, ratio=(.7, .2, .1),

                group_prefix=None)

IMAGE_SIZE = 256

CHANNELS = 3

from TensorFlow.Keras.preprocessing.image import ImageDataGenerator

train_datagen = ImageDataGenerator(

      rescale=1./255,

       rotation_range=60,

       horizontal_flip=True,

       vertical_flip=True,
```

```
    fill_mode='reflect'
)
train_generator = train_datagen.flow_from_directory(
        'datasetak71/train',
        target_size=(IMAGE_SIZE,IMAGE_SIZE),
        batch_size=32,
        class_mode="sparse"
)
train_generator.class_indices
class_names = list(train_generator.class_indices.keys())
class_names
count=0
for image_batch, label_batch in train_generator:
#     print(label_batch)
    print(image_batch[0])
    break
#     count+=1
#     if count>2:
#         break
validation_datagen = ImageDataGenerator(
         rescale=1./255,
```

```
        rotation_range=60,
        horizontal_flip=True,
        vertical_flip=True,
    fill_mode='reflect')
validation_generator = validation_datagen.flow_from_directory(
        'datasetak71/val',
        target_size=(IMAGE_SIZE,IMAGE_SIZE),
        batch_size=32,
        class_mode="sparse"
)
test_datagen = ImageDataGenerator(
        rescale=1./255,
        rotation_range=60,
        horizontal_flip=True,
        vertical_flip=True,
    fill_mode='reflect'
)
test_generator = test_datagen.flow_from_directory(
        'datasetak71/test',
        target_size=(IMAGE_SIZE,IMAGE_SIZE),
        batch_size=32,
```

```
    class_mode="sparse"
)
for image_batch, label_batch in test_generator:
  print(image_batch[0])
  break
from Keras.applications.vgg16 import VGG16
VGG_model = VGG16(
  include_top=False,
  weights="imagenet",
  input_tensor=None,
  input_shape=(IMAGE_SIZE,IMAGE_SIZE,3),
  classifier_activation="softmax",
)
for layer in VGG_model.layers:
  layer.trainable = False
VGG_model.summary()
from Keras. models import sequentially
from Keras. layers import Dense, Dropout, Flatten, Conv2D, MaxPool2D, BatchNormalization
model1= sequential
x = Flatten()(VGG_model.output)
```

```
pred=Dense(512,activation='relu')(x)

pred=Dropout(0.2)(x)

pred=Dense(7,activation='softmax')(x)

from Keras. models import Model

model = Model(inputs = VGG_model.input, outputs = pred)

model.summary()

model.compile(

    optimizer='adam',

    loss=tf.Keras. losses.SparseCategoricalCrossentropy(from_logits=False),

    metrics=['accuracy']

)

history = model.fit(

    train_generator,

    batch_size=32,

    validation_data=validation_generator,

    verbose=1,

    epochs=20,

)

scores = model.evaluate(test_generator)

acc = history.history['accuracy']

val_acc = history.history['val_accuracy']
```

```
loss = history.history['loss']

val_loss = history.history['val_loss']

EPOCHS = 20

figure(figsize=(8, 8))

subplot(1, 2, 1)

plot(range(EPOCHS), acc, label='Training Accuracy')

plot(range(EPOCHS), val_acc, label='Validation Accuracy')

.legend(loc='lower right')

title('Training and Validation Accuracy')

subplot(1, 2, 2)

plot(range(EPOCHS), loss, label='Training Loss')

plot(range(EPOCHS), val_loss, label='Validation Loss')

legend(loc='upper right')

title('Training and Validation Loss')

show()

import NumPy as np

for image_batch, label_batch in test_generator:

    first_image = image_batch[0]

    first_label = int(label_batch[0])

    print("first image to predict")

    pm show(first_image)
```

```
    print("actual label:",class_names[first_label])
    batch_prediction = model.predict(image_batch)
    print("predicted label:",class_names[np(batch_prediction[0])])
    break
def predict(model, img):
    img_array = tf.keras.preprocessing.image.img_to_array(images[i])
    img_array = tf.expand_dims(img_array, 0)
    predictions = model.predict(img_array)
    predicted_class = class_names[np.argmax(predictions[0])]
    confidence = round(100 * (np.max(predictions[0])), 2)
    return predicted_class, confidence
plt.figure(figsize=(15, 15))
for images, labels in test_generator:
    for I in range(6):
        ax = subplot(3, 3, i + 1)
        images[i])
        predicted_class, confidence = predict(model, images[i])
        actual_class = class_names[int(labels[i])]
       title(f"Actual: {actual_class},\n Predicted: {predicted_class}.\n Confidence: {confidence}%")
       axis("off")
```

APPENDICES

Appendix B

SECOND MODEL CODE

```
import TensorFlow

from TensorFlow.Keras import models, layers

import. as plt


import split folders

input_folder = 'C:/Users/SPACE/Desktop/Scabies-dataset-turns


splitfolders.ratio(input_folder, output="datasetak71",

        seed=42, ratio=(.7, .2, .1),

        group_prefix=None)


IMAGE_SIZE = 256

CHANNELS = 3


from tensorflow.keras.preprocessing.image import ImageDataGenerator


train_datagen = ImageDataGenerator(
```

```
      rescale=1./255,
       rotation_range=60,
       horizontal_flip=True,
       vertical_flip=True,
   fill_mode='reflect'
)
train_generator = train_datagen.flow_from_directory(
       'datasetak71/train',
       target_size=(IMAGE_SIZE,IMAGE_SIZE),
       batch_size=32,
       class_mode="sparse"
)

train_generator.class_indices

class_names = list(train_generator.class_indices.keys())
class_names

count=0
for image_batch, label_batch in train_generator:
#     print(label_batch)
```

```
    print(image_batch[0])
    break
#     count+=1
#     if count>2:
#         break

validation_datagen = ImageDataGenerator(
         rescale=1./255,
        rotation_range=60,
        horizontal_flip=True,
        vertical_flip=True,
    fill_mode='reflect')
validation_generator = validation_datagen.flow_from_directory(
        'datasetak71/val',
        target_size=(IMAGE_SIZE,IMAGE_SIZE),
        batch_size=32,
        class_mode="sparse"
)

test_datagen = ImageDataGenerator(
        rescale=1./255,
```

```
        rotation_range=60,
        horizontal_flip=True,
        vertical_flip=True,
    fill_mode='reflect'
)

test_generator = test_datagen.flow_from_directory(
        'datasetak71/test',
        target_size=(IMAGE_SIZE,IMAGE_SIZE),
        batch_size=32,
        class_mode="sparse"
)

for image_batch, label_batch in test_generator:
    print(image_batch[0])
    break

input_shape = (IMAGE_SIZE, IMAGE_SIZE, CHANNELS)
n_classes = 2

model = models.Sequential([
```

```
    layers.Conv2D(32, kernel_size = (3,3), activation='relu', input_shape=input_shape),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64,  kernel_size = (3,3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(128,  kernel_size = (3,3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(256, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(412, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(582, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Flatten(),
    layers.Dense(64, activation='relu'),
    layers.Dense(n_classes, activation='softmax'),
])

model.build(input_shape=input_shape)

model.summary()
```

```
model.compile(
    optimizer='adam',
    loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=False),
    metrics=['accuracy']
)

history = model.fit(
    train_generator,
    batch_size=32,
    validation_data=validation_generator,
    verbose=1,
    epochs=40,
)

scores = model.evaluate(test_generator)

acc = history.history['accuracy']
val_acc = history.history['val_accuracy']

loss = history.history['loss']
```

```
val_loss = history.history['val_loss']


EPOCHS = 40

plt.figure(figsize=(8, 8))

plt.subplot(1, 2, 1)

plt.plot(range(EPOCHS), acc, label='Training Accuracy')

plt.plot(range(EPOCHS), val_acc, label='Validation Accuracy')

plt.legend(loc='lower right')

plt.title('Training and Validation Accuracy')


plt.subplot(1, 2, 2)

plt.plot(range(EPOCHS), loss, label='Training Loss')

plt.plot(range(EPOCHS), val_loss, label='Validation Loss')

plt.legend(loc='upper right')

plt.title('Training and Validation Loss')

plt.show()


import NumPy as np


for image_batch, label_batch in test_generator:

    first_image = image_batch[0]
```

```
    first_label = int(label_batch[0])


    print("first image to predict")

    plt.imshow(first_image)

    print("actual label:",class_names[first_label])


    batch_prediction = model.predict(image_batch)

    print("predicted label:",class_names[np.argmax(batch_prediction[0])])


    break


def predict(model, img):

    img_array = tf.keras.preprocessing.image.img_to_array(images[i])

    img_array = tf.expand_dims(img_array, 0)


    predictions = model.predict(img_array)


    predicted_class = class_names[np.argmax(predictions[0])]

    confidence = round(100 * (np.max(predictions[0])), 2)

    return predicted_class, confidence
```

```
figure(figsize=(15, 15))
for images, labels in test_generator:
    for I in range(6):
        ax = subplot(3, 3, i + 1)
        (images[i])

        predicted_class, confidence = predict(model, images[i])
        actual_class = class_names[int(labels[i])]

        plt.title(f"Actual: {actual_class},\n Predicted: {predicted_class}.\n Confidence:
        {confidence}%")

       axis("off")
    break
```

APPENDICES

Appendix C

Turnitin Report

HUSAM ZENDAH MSC

INBOX | NOW VIEWING: NEW PAPERS ▾

Submit File

Online Grading Report | Edit assignment settings | Email non-submitters

AUTHOR	TITLE	SIMILARITY	GRADE	RESPONSE	FILE	PAPER ID	DATE
Husam Zendah	ABS 23062022	0%	--	--		1861729411	23-Jun-2022
Husam Zendah July	ABS_23072022	0%	--	--		1874085723	23-Jul-2022
Husam Zendah June	CH1_10062022	3%	--	--		1854142627	10-Jun-2022
Husam Zendah July	CH1_23072022	2%	--	--		1874085725	23-Jul-2022
Husam Zendah June 15	CH2_15062022	8%	--	--		1857813797	16-Jun-2022
Husam Zendah July	CH2_23072022	12%	--	--		1874085727	23-Jul-2022
Husam Zendah June 15	CH3_15062022	5%	--	--		1857813806	16-Jun-2022
Husam Zendah	CH3 25072022	7%	--	--		1874966551	25-Jul-2022
Husam Zendah June 13	CH4 13062022	3%	--	--		1855914450	13-Jun-2022
Husam Zendah July	CH4_23072022	2%	--	--		1874085737	23-Jul-2022
Husam J Zendah June	CH5_10062022	0%	--	--		1854143379	10-Jun-2022
Husam Zendah July	CH5_23072022	0%	--	--		1874085729	23-Jul-2022
Husam Zendah June 20	CONC 21062022	0%	--	--		1860605248	21-Jun-2022
Husam Z Zendah July	CONC_23072022	0%	--	--		1874090279	23-Jul-2022
Husam Z Zendah July	THESIS_23072022	8%	--	--		1874090280	23-Jul-2022

Thesis Title: USING OF CONVENTIONAL NEURAL NETWORK TO DIAGNOSE SCABIES BY DERMOSCOPY

Supervisor, Assoc. Prof. Dr. Kamil Dimililer.