

ESKİŞEHİR TEKNİK ÜNİVERSİTESİ

ESKISEHIR TECHNICAL UNIVERSITY

BİM447 Intoduction to Deep Learning

Detection Of Skin Diseases

FINAL REPORT

Alperen Burak Yeşil – 10129304460

Baran Bahtiyar – 19066449920

1. **Problem**

Early diagnosis of skin diseases has a significant impact on human health. When not diagnosed early, some skin conditions can lead to serious health problems. Therefore, the identification of skin diseases aims to improve patients' quality of life and is of vital importance. The use of technologies such as artificial intelligence, deep learning, and image processing can enhance the diagnosis and treatment processes. These technologies can assist doctors and provide faster and more accurate diagnoses, thus offering more effective treatment to patients. This helps prevent costly treatments in healthcare systems while enhancing overall public health and potentially saving lives, especially by detecting life-threatening conditions like skin cancer early. In conclusion, the identification of skin diseases is an important research and service area supported by the opportunities provided by modern medicine and technology, aimed at supporting healthy living and improving accessibility and effectiveness of healthcare services for people.

1. **Explanation of the Neural Network Model**

**Convolutional Neural Networks (CNN)**

Computer vision is one of the most exciting and rapidly advancing fields in computer science. Particularly in challenging tasks such as classification, object recognition, and segmentation, deep learning models like Convolutional Neural Networks (CNNs) have achieved remarkable success. CNNs are specifically designed to handle complexity in image data and are considered powerful tools for solving many problems in this field.

Unlike traditional neural networks, CNNs have a structure that allows them to recognize patterns in data effectively. This makes them ideal for tasks like image recognition and analysis. Many successful applications in image processing, such as autonomous driving, medical imaging, and security systems, involve effective utilization of CNNs.

The key to the success of CNNs lies in their architecture. These models consist of consecutive layers of convolution, activation, pooling, and fully connected layers. Convolutional layers perform the movement of filters to recognize features in the image data. Activation layers process the outputs of convolutional layers and add non-linearity. Pooling layers reduce dimensionality and preserve features from overlapping regions. Fully connected layers are used to assign features to classes.

In this article, we will delve into the fundamental components, architecture, and functions of CNNs in more detail. We will also discuss how CNNs are trained and how they are used in various practical applications. Ultimately, by highlighting the significance and future potential of CNNs in the field of image processing, we aim to provide readers with a deeper understanding of this powerful deep learning model.

**Convolutional Layers:**

Convolutional layers are one of the fundamental components of a Convolutional Neural Network (CNN) and are commonly used in image processing tasks. These layers perform the movement of filters (kernels) over the input data to detect features. For example, they can be used to identify edges, shapes, or other specific features in an image.

Function:

The main function of convolutional layers is to identify local and overlapping regions of features on the input data. This helps in detecting objects or features by identifying patterns and structures in different regions of the input data.

Convolution Process:

The convolution process occurs by applying a filter (kernel) or weight matrix to the input data by moving it across the data. During this process, the filter is slid over each region of the input data in an overlapping manner, and an output is produced for each region. These outputs represent the regions where the filters detect specific features.

Weights and Feature Maps:

Convolutional layers contain a set of weights for each filter. These weights represent the values of the filters (kernels) and are used to identify specific features. When each filter is applied, a feature map is created on the input data. Feature maps represent the outputs obtained from applying each filter on the input data. Each feature map represents a specific feature detected by the filter.

kare, dikdörtgen, çizgi, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, kare, ekran görüntüsü, dikdörtgen içeren bir resim

Açıklama otomatik olarak oluşturulduConvolutional layers increase the complexity of the model while extracting specific features from the input data. This enables the identification of more complex features in higher layers. These features can ultimately be used for classification or other tasks.

ekran görüntüsü, kare, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Original Image Filter

**Activation Layers:**

Activation layers apply an activation function to the outputs of convolutional layers. These layers are essential to enhance the learning capability of the model and make it resistant to overfitting. The most commonly used activation function is the Rectified Linear Unit (ReLU) function. ReLU sets negative inputs to zero while retaining positive inputs.

Function:

The primary function of activation layers is to introduce non-linearity into the network, allowing it to learn complex patterns and relationships in the data. By applying an activation function to the outputs of convolutional layers, activation layers enable the network to model more intricate functions and make more accurate predictions.

ReLU Activation Function:

ReLU is widely preferred due to its simplicity and effectiveness. It helps alleviate the vanishing gradient problem by preventing gradients from becoming too small during backpropagation. Additionally, ReLU accelerates convergence during training by enabling faster learning in the network.

Other Activation Functions:

While ReLU is the most commonly used activation function, other functions such as Sigmoid and Hyperbolic Tangent (Tanh) are also used in certain scenarios. Sigmoid function squashes the output to the range [0, 1], making it suitable for binary classification tasks. Tanh function squashes the output to the range [-1, 1], enabling the model to capture negative as well as positive features in the data.

Activation layers play a crucial role in deep learning models by introducing non-linearity and enabling the network to learn complex representations from the data. Proper selection of activation functions is essential for achieving optimal performance and training stability in neural networks.

siyah beyaz, metin, beyaz, monokrom, tek renkli içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Pooling Layers:**

Pooling layers are used to reduce the size of the network and computational load. These layers decrease the dimensions of feature maps, highlighting overlapping features. Typically, techniques like max pooling or average pooling are employed. Max pooling reduces the size by taking the maximum value from each region, while average pooling computes the average value from each region.

Function:

The primary function of pooling layers is to reduce the spatial dimensions of the input volume, thereby reducing the number of parameters and computational load in the network. By reducing the size of feature maps, pooling layers help in controlling overfitting and improving the generalization capability of the model.

Max Pooling:

In max pooling, a window moves over the input feature map, and the maximum value within each window is selected to represent that region. This process effectively down samples the feature maps while preserving the most important information, such as the presence of certain features.

Average Pooling:

Contrary to max pooling, average pooling computes the average value of each window instead of selecting the maximum. This technique also reduces the size of feature maps but may result in a smoother representation of the features compared to max pooling.

Strides and Padding:

Pooling layers can also incorporate parameters such as stride and padding to control the down sampling process. Stride determines the step size of the window movement, while padding adds additional border pixels to the input feature map, ensuring that all pixels are considered during pooling.

Pooling layers are essential components in CNNs for reducing spatial dimensions and controlling computational complexity. By down sampling feature maps, pooling layers help in extracting the most salient information from the data, facilitating efficient learning and inference processes. However, many people do not prefer to use this layer. Instead, a larger Stride (filter shifting process) is preferred in the Convolutional layer. They also completely omit the pooling layer in more generative models such as variational autoencoders (VAEs) or generative adversarial networks (GANs).

ekran görüntüsü, kare, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Flattening Layer:**

The flattening layer is commonly used in the structure of Convolutional Neural Networks (CNNs). Its main task is to prepare the data for the input of the fully connected layer, which is typically the final and most critical layer. Neural networks generally take input data as a one-dimensional array. However, the outputs from previous layers are often 2D or 3D matrices. The flattening layer converts these matrices into a one-dimensional array, making them suitable for the fully connected layer.

Function:

The primary function of the flattening layer is to flatten the outputs from Convolutional and Pooling layers into a one-dimensional array. This process combines the feature maps obtained from previous layers into a single vector. Consequently, these features can be directly fed into the input of the fully connected layer, preparing them for the classification process.

Application:

The flattening layer typically appears in the final stages of a CNN, where feature extraction is completed, and the classification process is to be performed. While the outputs of Convolutional and Pooling layers are used for feature detection, the flattening layer prepares these features for the classification process.

The flattening layer plays a critical role in training CNNs and predicting outcomes. By simplifying complex structures from previous layers, this layer provides a suitable input for the fully connected layer, which is used for the classification process.

diyagram içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Fully Connected Layers:**

Fully connected layers form the final stage of a Convolutional Neural Network (CNN) and are typically used for the classification process. These layers take the outputs of convolutional and pooling layers and pass them through the classification process. It is the point where features are directly assigned to classes, and class labels are predicted.

Function:

The primary function of fully connected layers is to prepare the data from previous layers, where features are extracted, for classification. These layers take feature vectors from previous layers and process them into a format that can be handled by the classifier algorithm. Consequently, each feature vector is assigned to a specific class, leading to the final classification.

SoftMax Activation Function:

Typically, SoftMax activation function is used at the output of fully connected layers. The SoftMax function creates a probability distribution for each class, allowing for competitive decision-making among classes. Consequently, the predicted probability of each class is obtained, and the class with the highest probability is predicted.

Fully connected layers are the part of CNN that comes after the feature extraction stage and performs the classification process. These layers play a critical role in training CNNs and predicting outcomes.

**Output Layer:**

The output layer is the layer that produces the output of a model. In classification problems, class probabilities are typically obtained using the SoftMax activation function. This function creates a probability distribution for each class and predicts the class with the highest probability. In regression problems, direct predictions are typically generated, so no activation function is used.

**Learning Process:**

Convolutional Neural Networks (CNNs) are trained using the backpropagation algorithm. In this process, a loss function is used to calculate the difference between the model's predictions and the actual labels. Then, the parameters of the network are updated based on this difference. This is usually done using an optimization algorithm such as Stochastic Gradient Descent (SGD), Adam, or RMSProp.

**Architectures of Convolutional Neural Networks**

A simple way to build a CNN is to stack several Convolutional Layers one after the other and add a ReLU layer after each one. And after that, Pooling layer(s) and Flattening layer should be added. Then, as many Fully Connected layers as ReLU layers are added. Keep in mind that the last layer of a CNN should be a Fully Connected layer. It can be defined as follows:

Input Image -> [[Conv -> ReLU]\*N -> Pool?]\*M -> Flattening -> [FC -> ReLU]\*K -> FC

Here are some common ConvNet names:

LeNet - This network is considered as the first successful application of Convolutional Networks. It was developed by Yann LeCun in the 1990s and was used to read zip codes, simple digits, etc.

AlexNet - This network was introduced in 2012 at the ImageNet ILSVRC challenge. It performed significantly better than other networks.

GoogLeNet - This network won the ILSVRC 2014 competition. They used average pooling layers to significantly reduce the number of parameters in the network.

VGGNet - This network has proven how important the depth of the network is. It contains 16 convolutional layers.

1. **Description of the dataset**

The HAM10000 dataset is specifically curated for the training of neural networks used in the automated diagnosis of skin lesions. This dataset is designed particularly for tasks such as classification and identification of dermoscopic images.

The dataset contains dermoscopic images obtained from different populations and various types of skin lesions. Images may have been captured under different resolutions and lighting conditions. However, each image's lesion is labeled with tags representing various categories of skin lesions.

The labels in the dataset include:

- Actinic keratoses and intraepithelial carcinoma / Bowen's disease (akiec)

- Basal cell carcinoma (bcc)

- Benign keratosis-like lesions (solar lentigines / seborrheic keratoses and lichen-planus like keratoses, bkl)

- Dermatofibroma (df)

- Melanoma (mel)

- Melanocytic nevi (nv)

- Vascular lesions (angiomas, angiokeratomas, pyogenic granulomas and hemorrhage, vasc)

Each image is paired with a label indicating the type of lesion it represents. These labels are used to accurately classify and identify the images.

The metadata provided as part of the dataset includes information about each image, along with their labels and relevant meta-information. This meta-information includes details such as how each image was obtained, which modalities were used for storage, and the basis of the labels.

All this information is designed to support the training and evaluation of neural networks used in the automatic diagnosis of skin lesions. The dataset provides a valuable resource to support research in this field.

The features of each example are as follows:

- lesion\_id: Unique identifier of the lesion

- image\_id: Unique identifier of the image associated with the lesion

- dx: Diagnosis code of the skin disease

- dx\_type: Diagnosis method (such as histo)

- age: Age of the patient

- sex: Gender of the patient

- localization: Localization of the lesion on the body

1. **Results**

Accuracy: Our model has been quite successful in accurately classifying the skin lesions in our dataset. It achieved high accuracy during the training process and also delivered satisfactory results on the test dataset.

Disease Detection: Our model successfully detected various skin lesions. By analyzing the given images, it accurately identified the presence of specific diseases.

Areas for Improvement: There are some areas where our model can be improved. For example, to achieve greater sensitivity in detecting certain diseases, the model could be trained further, or the dataset could be made more balanced.

Real-World Applications: The accuracy and reliability of our model make it feasible for use in real-world applications. It could be utilized as an assistive tool for dermatologists or healthcare professionals.

Continuous Improvement: It is crucial to continuously improve and update our model. More data, better architectures, and fine-tuning of hyperparameters can enhance the model's performance and enable it to cover a broader range of diseases.

metin, ekran görüntüsü, çizgi, öykü gelişim çizgisi; kumpas; grafiğini çıkarma içeren bir resim

Açıklama otomatik olarak oluşturulduIn conclusion, the successes of our model are significant, and it can play an important role in the automatic diagnosis of skin lesions. However, it must be continually developed and improved to ensure it can be used most effectively in real-world applications.

metin, ekran görüntüsü, diyagram, öykü gelişim çizgisi; kumpas; grafiğini çıkarma içeren bir resim

Açıklama otomatik olarak oluşturuldu

In conclusion, the successes of our model are significant, and it can play an important role in the automatic diagnosis of skin lesions. However, it must be continually developed and improved to ensure it can be used most effectively in real-world applicationsmetin, ekran görüntüsü, öykü gelişim çizgisi; kumpas; grafiğini çıkarma, diyagram içeren bir resim

Açıklama otomatik olarak oluşturuldu

1. **Dataset**

|  |  |
| --- | --- |
| **lesion\_id, image\_id, dx, dx\_type, age, sex, localization** |  |
| HAM\_0000118,ISIC\_0027419,bkl,histo,80.0,male,scalp | |
| HAM\_0000118,ISIC\_0025030,bkl,histo,80.0,male,scalp | |
| HAM\_0002730,ISIC\_0026769,bkl,histo,80.0,male,scalp | |
| HAM\_0002730,ISIC\_0025661,bkl,histo,80.0,male,scalp | |
| HAM\_0001466,ISIC\_0031633,bkl,histo,75.0,male,ear |  |
| HAM\_0001466,ISIC\_0027850,bkl,histo,75.0,male,ear |  |
| HAM\_0002761,ISIC\_0029176,bkl,histo,60.0,male,face | |
| HAM\_0002761,ISIC\_0029068,bkl,histo,60.0,male,face | |
| HAM\_0005132,ISIC\_0025837,bkl,histo,70.0,female,back | |
| HAM\_0005132,ISIC\_0025209,bkl,histo,70.0,female,back | |
| HAM\_0001396,ISIC\_0025276,bkl,histo,55.0,female,trunk | |
| HAM\_0004234,ISIC\_0029396,bkl,histo,85.0,female,chest | |
| HAM\_0004234,ISIC\_0025984,bkl,histo,85.0,female,chest | |
| HAM\_0001949,ISIC\_0025767,bkl,histo,70.0,male,trunk | |
| HAM\_0001949,ISIC\_0032417,bkl,histo,70.0,male,trunk | |
| HAM\_0007207,ISIC\_0031326,bkl,histo,65.0,male,back | |
| HAM\_0001601,ISIC\_0025915,bkl,histo,75.0,male,upper extremity | |
| HAM\_0001601,ISIC\_0031029,bkl,histo,75.0,male,upper extremity | |
| HAM\_0007571,ISIC\_0029836,bkl,histo,70.0,male,chest | |
| HAM\_0007571,ISIC\_0032129,bkl,histo,70.0,male,chest | |
| HAM\_0006071,ISIC\_0032343,bkl,histo,70.0,female,face | |
| HAM\_0003301,ISIC\_0025033,bkl,histo,60.0,male,back | |
| HAM\_0003301,ISIC\_0027310,bkl,histo,60.0,male,back | |
| HAM\_0004884,ISIC\_0032128,bkl,histo,75.0,male,upper extremity | |
| HAM\_0004884,ISIC\_0025937,bkl,histo,75.0,male,upper extremity | |
| HAM\_0002521,ISIC\_0027828,bkl,histo,40.0,male,upper extremity | |
| HAM\_0002521,ISIC\_0029291,bkl,histo,40.0,male,upper extremity | |
| HAM\_0006574,ISIC\_0030698,bkl,histo,40.0,male,back | |
| HAM\_0006574,ISIC\_0025567,bkl,histo,40.0,male,back | |
| HAM\_0001480,ISIC\_0031753,bkl,histo,70.0,male,abdomen | |
| HAM\_0001480,ISIC\_0026835,bkl,histo,70.0,male,abdomen | |
| HAM\_0005772,ISIC\_0031159,bkl,histo,60.0,female,face | |
| HAM\_0005772,ISIC\_0031017,bkl,histo,60.0,female,face | |
| HAM\_0005612,ISIC\_0024981,bkl,histo,80.0,male,scalp | |
| HAM\_0005388,ISIC\_0027815,bkl,histo,80.0,male,chest | |
| HAM\_0000351,ISIC\_0024324,bkl,histo,85.0,male,back | |
| HAM\_0000351,ISIC\_0029559,bkl,histo,85.0,male,back | |
| HAM\_0003847,ISIC\_0030661,bkl,histo,85.0,male,upper extremity | |
| HAM\_0003847,ISIC\_0027053,bkl,histo,85.0,male,upper extremity | |
| HAM\_0003847,ISIC\_0028560,bkl,histo,85.0,male,upper extremity | |
| HAM\_0003847,ISIC\_0031650,bkl,histo,85.0,male,upper extremity | |
| HAM\_0000164,ISIC\_0029161,bkl,histo,60.0,male,chest | |
| HAM\_0000164,ISIC\_0026273,bkl,histo,60.0,male,chest | |
| HAM\_0007409,ISIC\_0025076,bkl,histo,50.0,male,upper extremity | |
| HAM\_0007409,ISIC\_0029687,bkl,histo,50.0,male,upper extremity | |
| HAM\_0007409,ISIC\_0025642,bkl,histo,50.0,male,upper extremity | |
| HAM\_0002299,ISIC\_0025819,bkl,histo,75.0,female,face | |
| HAM\_0002299,ISIC\_0032013,bkl,histo,75.0,female,face | |
| HAM\_0007010,ISIC\_0031691,bkl,histo,40.0,male,trunk | |
| HAM\_0007010,ISIC\_0025419,bkl,histo,40.0,male,trunk | |
| HAM\_0003670,ISIC\_0030105,bkl,histo,80.0,female,unknown | |
| HAM\_0007125,ISIC\_0025016,bkl,histo,75.0,male,back | |
| HAM\_0007125,ISIC\_0029147,bkl,histo,75.0,male,back | |
| HAM\_0001221,ISIC\_0029301,bkl,histo,45.0,male,upper extremity | |
| HAM\_0001221,ISIC\_0026637,bkl,histo,45.0,male,upper extremity | |
| HAM\_0001983,ISIC\_0030377,bkl,histo,70.0,male,back | |
| HAM\_0003569,ISIC\_0027960,bkl,histo,75.0,male,unknown | |
| HAM\_0003569,ISIC\_0026955,bkl,histo,75.0,male,unknown | |
| HAM\_0000700,ISIC\_0028052,bkl,histo,60.0,male,face | |
| HAM\_0000728,ISIC\_0025286,bkl,histo,50.0,male,lower extremity | |
| HAM\_0003021,ISIC\_0031468,bkl,histo,75.0,female,face | |
| HAM\_0003021,ISIC\_0030926,bkl,histo,75.0,female,face | |
| HAM\_0000959,ISIC\_0029288,bkl,histo,75.0,female,face | |
| HAM\_0000959,ISIC\_0031008,bkl,histo,75.0,female,face | |
| HAM\_0001751,ISIC\_0024698,nv,consensus,70.0,male,face | |
| HAM\_0004569,ISIC\_0031495,bkl,histo,40.0,male,upper extremity | |
| HAM\_0004569,ISIC\_0026104,bkl,histo,40.0,male,upper extremity | |
| HAM\_0004641,ISIC\_0025099,bkl,histo,60.0,male,abdomen | |
| HAM\_0004641,ISIC\_0031485,bkl,histo,60.0,male,abdomen | |
| HAM\_0005801,ISIC\_0029413,bkl,histo,70.0,female,abdomen | |
| HAM\_0005801,ISIC\_0029576,bkl,histo,70.0,female,abdomen | |
| HAM\_0004341,ISIC\_0031967,bkl,histo,70.0,female,scalp | |
| HAM\_0004341,ISIC\_0031584,bkl,histo,70.0,female,scalp | |
| HAM\_0000907,ISIC\_0025140,bkl,histo,75.0,male,ear |  |
| HAM\_0000907,ISIC\_0025554,bkl,histo,75.0,male,ear |  |
| HAM\_0001469,ISIC\_0029289,bkl,histo,60.0,female,trunk | |
| HAM\_0001469,ISIC\_0029912,bkl,histo,60.0,female,trunk | |
| HAM\_0001728,ISIC\_0033539,bkl,histo,60.0,male,back | |
| HAM\_0003021,ISIC\_0032283,bkl,histo,75.0,female,face | |
| HAM\_0003021,ISIC\_0030005,bkl,histo,75.0,female,face | |
| HAM\_0000959,ISIC\_0030189,bkl,histo,75.0,female,face | |
| HAM\_0000959,ISIC\_0026532,bkl,histo,75.0,female,face | |
| HAM\_0001773,ISIC\_0024832,bkl,histo,40.0,female,face | |
| HAM\_0001773,ISIC\_0026958,bkl,histo,40.0,female,face | |
| HAM\_0002127,ISIC\_0030768,bkl,histo,40.0,male,face | |
| HAM\_0002127,ISIC\_0029837,bkl,histo,40.0,male,face | |
| HAM\_0002092,ISIC\_0031624,bkl,histo,35.0,female,lower extremity | |
| HAM\_0002092,ISIC\_0025510,bkl,histo,35.0,female,lower extremity | |
| HAM\_0005075,ISIC\_0030607,bkl,histo,80.0,male,upper extremity | |
| HAM\_0005075,ISIC\_0029060,bkl,histo,80.0,male,upper extremity | |
| HAM\_0002921,ISIC\_0029308,bkl,histo,60.0,female,face | |
| HAM\_0003410,ISIC\_0024635,bkl,histo,60.0,female,face | |
| HAM\_0003410,ISIC\_0029425,bkl,histo,60.0,female,face | |
| HAM\_0004852,ISIC\_0028774,bkl,histo,65.0,male,face | |
| HAM\_0004852,ISIC\_0030565,bkl,histo,65.0,male,face | |
| HAM\_0000746,ISIC\_0027023,bkl,histo,60.0,male,face | |
| HAM\_0001473,ISIC\_0029022,bkl,histo,70.0,male,face | |
| HAM\_0003007,ISIC\_0025388,bkl,histo,40.0,female,abdomen | |
| HAM\_0003007,ISIC\_0028080,bkl,histo,40.0,female,abdomen | |
| HAM\_0002957,ISIC\_0026153,bkl,histo,70.0,male,back | |