Radyografik Görüntüler Kullanılarak Vücut Anomalisi Tespiti

**Zahra Mohamed Kamel Harira**

**YÜKSEK LİSANS TEZİ**

Bilgisayar Mühendisliği Anabilim Dalı

Haziran 2022

Detection of Body Anomalies Using Radiographic Images

**Zahra Mohamed Kamel Harira**

**MASTER OF SCIENCE THESIS**

Department of Computer Engineering

June 2022

Detection of Body Anomalies Using Radiographic Images

A thesis submitted to the Eskisehir Osmangazi University Graduate School of Natural and Applied Sciences in partial fulfillment of the requirements for the degree of Master of Science in Discipline of Computer Software of the Department of Computer Engineering

by

**Zahra Mohamed Kamel Harira**

Supervisor: **Assist. Prof. Dr. Efnan Şora Günal**

June 2022

**ETHICAL STATEMENT**

I hereby declare that this thesis study titled “Detection of Body Anomalies Using Medical Images” has been prepared in accordance with the thesis writing rules of Eskişehir Osmangazi University Graduate School of Natural and Applied Sciences under academic consultancy of my supervisor **Assist. Prof. Dr. Efnan Şora Günal**. I hereby declare that the work presented in this thesis is original. I also declare that, I have respected scientific ethical principles and rules in all stages of my thesis study, all information and data presented in this thesis have been obtained within the scope of scientific and academic ethical principles and rules, all materials used in this thesis which are not original to this work have been fully cited and referenced, and all knowledge, documents and results have been presented in accordance with scientific ethical principles and rules 08/06/2022

**Zahra Mohamed Kamel Harira**

# ÖZET

Kas-iskelet sistemi yaralanmaları vücut yapılarının bozukluklarıdır. Bu bozukluklar kaslarda, kemiklerde, eklemlerde, sinirlerde, kemiklerde ve diğerlerinde bulunabilir, diz ve kemiklerde anormalliklere, sırt ağrısına ve diğer birçok eklem veya kas problemine neden olabilir. Kas-iskelet yaralanmaları, acil servise yılda 30 milyona varan plansız ziyaretlerin önde gelen nedenleri arasındadır. Bu çalışmanın temel amacı, X-ışını görüntülerinden vücut anomalilerini tespit etmek için yapay zeka temelli sınıflandırma modelleri geliştirmektir. Bu amaçla, ImageNet veri seti ile önceden eğitilmiş beş farklı derin sinir ağı modeli, transfer öğrenme yaklaşımı kullanılarak büyük bir X-ray görüntü veri seti ile yeniden eğitilmiştir. Bu veri seti, omuz, humerus, dirsek, önkol, bilek, el ve parmak dahil olmak üzere üst ekstremiteye ait 14.656 kas-iskelet çalışmasını içermektedir. Çalışmada, bir görüntü ön işleme yöntemi olan Kontrast Sınırlı Adaptif Histogram Eşitlemenin (CLAHE) sınıflandırma performansına katkısı da araştırılmıştır. Ayrıca, performansı daha da iyileştirmek amacıyla topluluk öğrenimi yaklaşımı da kullanılmıştır. Kapsamlı deneyler, vücut anomalilerinin %86'ya varan doğrulukla tespit edilebileceğini ortaya koymaktadır. Gelecek vaat eden bu sınıflandırma performansı göz önüne alındığında, doktorlara yardımcı olmak için, önerilen modeller kullanılarak X-ışını görüntülerini otomatik olarak sınıflandıran gerçek zamanlı bir mobil uygulama geliştirilebilir.

**Anahtar Kelimeler:** Yapay Zeka, Medikal Görüntüleme, Anomali Tespiti, Derin Öğrenme, Transfer Öğrenme, Hiperbant Optimizasyonu, Topluluk Öğrenimi.

## SUMMARY

Musculoskeletal injuries are disorders of body structures. These disorders can be found in muscles, bones, joints, nerves, bones, and others, causing abnormalities in the knee and bones, back pain, and many other joint or muscle problems. Musculoskeletal injuries are among the leading causes of unplanned visits to the emergency department, numbering up to 30 million a year. The main goal of this study is to develop artificial intelligence-based classification models to detect body abnormalities from X-ray images. For this purpose, five different deep neural network models pre-trained with the ImageNet dataset are retrained with a large X-ray image dataset using the transfer learning approach. The dataset contains 14,656 musculoskeletal studies of the upper extremity including the shoulder, humerus, elbow, forearm, wrist, hand, and finger. Also, the contribution of the image preprocessing method, Contrast Limited Adaptive Histogram Equalization (CLAHE), to the classification performance is investigated. Furthermore, ensemble learning is used to further improve performance. Extensive experiments reveal that body abnormalities can be detected with an accuracy of up to 86%. Considering the promising classification performance, a real-time mobile application automatically classifying X-ray images can be developed using the proposed models to assist physicians.

**Keywords:** Artificial Intelligence, Medical Imaging, Abnormality Detection, Deep Learning, Transfer Learning, Hyperband Optimization, Ensemble Learning.

# LIST OF CONTENTS

**Page**

## LIST OF CONTENTS (continued)

**Page**

# LIST OF FIGURES

**Figure** **Page**

# LIST OF TABLES

**Table** **Page**

# LIST OF ABBREVIATIONS AND SYMBOLS

| **Symbols** | **Description** |
|---|---|
| $r_k$ | input intensity |
| $s_k$ | processed intensity |
| k | intensity range |
| $n_j$ | frequency of intensity j |
| n | sum of all frequencies |

| **Abbreviations** | **Description** |
|---|---|
| CT | Computed Tomography |
| MRI | Magnetic Resonance Imaging |
| CNN | Convolutional neural network |
| CLAHE | Contrast Limited Adaptive Histogram Equalization |
| DL | Deep learning |
| AI | Artificial intelligence |
| ReLU | The Rectified Linear Unit |
| FC | Fully connected |
| TL | Transfer learning |
| FE | Feature extraction |
| FT | Fine-tuning |
| SPECT | Single Photon Emission Computed Tomography |

# 1. INTRODUCTION AND PURPOSE

Over recent years, we have witnessed the rapid technical developments in medical imaging technology and how Artificial Intelligence revolutionized many types of images such as X-ray images, ultrasound, SPECT, MRI, and others.

Nowadays, medical image analysis has become a very active research area, having many researchers inspecting health-related data. Approaches such as deep learning, transfer learning, machine learning, and data mining are used to extract features from images to detect and predict different diseases (Abumalloh et al., 2021). Additionally, Medical imaging is a fast-developing field that provides accurate results on computer systems (Toğaçar et al., 2019). The image processing technique is used in almost all sectors of healthcare (Bhattacharya et al., 2020), one of which is the Musculoskeletal sector.

Musculoskeletal injuries are disorders of body structures. These disorders can be found in muscles, bones, joints, nerves, bones, and others, causing abnormalities in the knee and bones, back pain, and many other joint or muscle problems (He et al., 2021). According to a study on the Global Burden of Disease, Musculoskeletal disorders are affecting nearly 2 billion people in the world, which makes it to have a great effect on the health of the people causing death and disability (BMUS,2015). Musculoskeletal injuries are among the leading causes of unplanned visits to the emergency department (Ajmera et al.,2021), numbering up to 30 million a year (He et al., 2021). They can be caused by sports injuries, genetics, and work accidents such as fractions which cause long-term pain and even disability (BMUS,2015). Musculoskeletal conditions lead to absence from work, and early or forced retirement due to limited mobility (Saif et al., 2019).

There are different types of medical imaging tools for detecting abnormalities. Some of the commonly used examples are X-ray images, ultrasound (US), Magnetic Resonance Imaging (MRI), and Computed Tomography (CT) (Anu et al.,2015). X-ray images of the musculoskeletal system are basically used for their speed and easiness in viewing bone abnormalities such as fractures, injuries, and joint disorders (He et al., 2021). However, years of experience and knowledge are needed for reading musculoskeletal X-rays in the right way. The lack of medical professionals is an essential problem in many countries. Besides that, there are many young specialists, who do not have sufficient experience to read correctly the X-ray images and consequently cannot detect complicated injuries (Solovoya et al.,2020). All these are common reasons for misdiagnosing.

Another problem is that experienced radiologists may not be available in busy medical centers or clinicians may not have enough time to interpret the musculoskeletal abnormality, which may result in a misdiagnosis. Considering that the cases are increasing every single day, the usual diagnostic process needs to be improved. As a solution, Deep Learning approaches are used in many studies that showed the effectiveness of these techniques applied to the images (Arias-Garzón et al.,2021).

There among the open-access medical image datasets is the large MURA dataset, published in Jan 2018 by Stanford ML Group (Solovoya et al.,2020). It contains 14,863 studies with a total of 40,005 radiographic images of the upper extremity including images of the elbow, shoulder, hand, humerus, finger, forearm, and wrist (Rajpurkar et al.,2018).

**Purpose of the Thesis**

This study aims to use deep learning methods to provide radiologists and medical professionals with a software tool to review their interpretation of the disease in the diagnostic process. In addition, in most developing countries, due to the high cost, people do not have access to radiologists. Therefore this tool can help healthcare professionals interpret X-ray images. Besides that, it can be very helpful in the emergency department. For this purpose, musculoskeletal X-ray images shared by Stanford ML Group were used. In this dataset, 40,005 X-ray images are divided into two classes and labeled as Normal and Abnormal.

# 2. LITERATURE REVIEW

## 2.1. Related Works

Rajpurkar et al. in their study (2018), introduced the publicly available MURA dataset, a dataset of musculoskeletal radiographs of the upper extremity. A Dense Convolutional Neural Network with 169 layers was used for classifying the normality and abnormality of the radiographs. After applying image resizing and image augmentation the authors' model achieved an AUROC of 0.929, a sensitivity of 0.815, and a specificity of 0.887. In addition, the model achieved better results in classifying images of the wrist and finger compared to the performance of the best radiologist.

Saif et al. (2019) present the development of a model using Capsule Network architecture for detecting and classifying musculoskeletal X-ray images. Two algorithms Naturalness Image Quality Evaluator (NIQE) and Blind Image Spatial Quality Evaluator (BRISQUE) were used to calculate the quality score of an image with computational efficiency after the model is trained, ending up with resizing the images to 224 × 224. (Saif et al., 2019). To see how successful the model is, the Cohen's kappa score obtained from it was compared with this obtained using DenseNet. Considering that half of the data was used for training and validation and the other half for testing, the proposed model gained a Cohen's kappa score 10 % better than the DenseNet proposed by Rajpurkar et al.

The use of two different deep learning models is presented in the work of Choudhary et al. (2020). The first one VGG16, formed using 5 convolution blocks and using the sigmoid activation function in the output layers, achieved 76% of accuracy, a specificity of 0.827, a sensitivity of 0.701, and a Cohen's Kappa score reaching 0.532. Compared to the results from Rajpurkar's work (Rajpurkar et al., 2018), taken as a baseline, VGG16 outperformed the baseline in the case of the finger study (Choudhary et al. 2020). The second used model was DenseNet169, containing a total of 5 dense blocks, which obtained a Kappa score of 0.616. At the end of the study, it was concluded that DenseNet169 achieved better overall performance, taking into account the outperforming of the baseline in the elbow, finger, and humerus studies.

DenseNet169 was also used in the experimental study of Solovoya et al. (2020). As a first step, the region of interest (RoI) in the images was found for the classification. Using Dense Convolutional Network architecture the authors managed to achieve results for the Cohen’s kappa score for the wrist, hand, and shoulder as 0.942, 0.862, and 0.735 respectively, which were better than those published on the website of the challenge.

In the study of He et al. (2021), they evaluated five deep learning approaches: three widely used models (ConvNet, ResNet, and DenseNet), a meta-learner with ResNet, and DenseNet, and the authors’ proposed calibrated ensemble learner. They concluded that comparing the success of the three individual models, ResNet and DenseNet similarly achieved better results than ConvNet. However, comparing them to the ensemble learners showed the outperforming of the customized models. The models, having the two best evaluation metrics achieved from the validation, were selected for training the calibrated ensemble learner. DenseNet was chosen for forearm and elbow studies, and for the other five categories, ResNet was selected (He et al., 2021). As a result of this study, it is proven that the calibrated model achieves the highest performance in the overall category.

On the other hand, Chada (2019) decided to experiment only on humerus and finger subsets, as they had the lowest agreement rating with the radiologist according to Rajpurkar’s study. Three deep convolutional neural network architectures, DenseNet169, DenseNet201, and InceptionResNetV2, were used showing accuracies greater than 80%. DenseNet201 with a kappa score (0.764) performed the best on the humerus test set and InceptionResNetV2 with a Cohen’s kappa score of 0.555 on the finger set. Both models provided better results compared to Rajpurkar’s study on humerus and finger sets.

## 2.2. X-ray Imaging

X-rays were first discovered in 1895 by German Physics Professor Wilhelm Conrad Röntgen while working at the University of Würzburg. The discovery of X-rays first began when Röntgen noticed a faint green glow from the screen while investigating cathode rays from a Crookes tube. He realized some invisible rays coming from the tube were passing through the cardboard and also through books and papers on his desk. With this discovery, the first X-ray devices started to be manufactured in the same year.

Compared to visible light, X-rays have more energy and a smaller wavelength as shown in Fig. 2.1 which means they can easily pass through human skin and soft tissues in the body but not through hard tissues such as bone (Arslan, T., 2015). X-ray devices were invented by taking advantage of this property. X-ray is being applied to the area to be imaged and a photographic film or detector is placed behind the applied area, so when X-rays pass through soft tissues, they leave traces, and shaded sections are formed on the plates since they cannot pass through hard tissues. In this way, problems in hard tissues are detected.

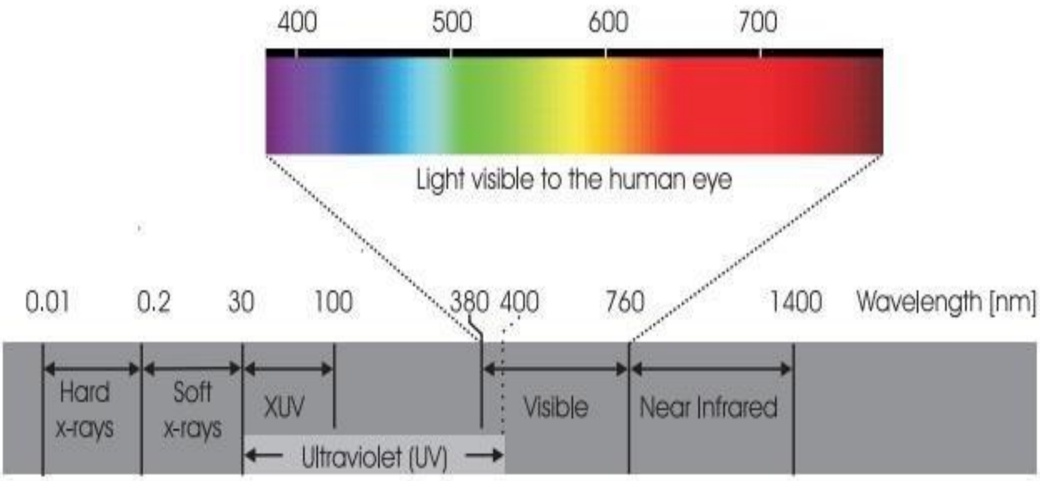


Figure 2.1. X-rays in the electromagnetic spectrum (Elton, 1990)

X-ray is an imaging technique that is very fast and does not make the patient feel pain. There are different types of X-rays used to detect different diseases. An example of a musculoskeletal X-ray is given in Figure 2.2.


Figure 2.2. Musculoskeletal X-Ray images from MURA datatset

# 3. METHODS

## 3.1. Image Preprocessing

Image pre-processing is normally used to remove undesired noise from images, highlight needed details from the images, and others.

### 3.1.1. Contrast Limited Adaptive Histogram Equalization (CLAHE)

One of the widely used image preprocessing methods is histogram equalization. The histogram shows us the representation of the intensity distribution of an image graphically Fig. 3.1. To improve the contrast in images, the histogram equalization technique is used. The main idea here is for areas with lower contrast to have higher contrast. This is achieved by spreading out the frequencies using the following formula (Eq. 3.1):

$$s_k = T(r_k) = (L-1)\Sigma\, p_r(r_j) = (L-1)\,\Sigma\, n_j / n \tag{3.1}$$

where:

$r_k$: is the input intensity

$s_k$: processed intensity

$k$: the intensity range (0,1,2,..., L-1)

$n_j$: the frequency of intensity j

$n$: the sum of all frequencies

The Contrast Limited Adaptive Histogram Equalization CLAHE depends on a transformation function to transform each pixel which makes it different from the normal histogram equalization (Muniyappan et al., 2013). In Fig. 3.2 the original image is shown before and after applying CLAHE.

### 3.1.2. Data Augmentation

The achievement of a deep convolutional neural network relies mainly on the size of the dataset (Shorten et al., 2019). The bigger and the more various the dataset is the better results are achieved. However, collecting data is time-consuming and can be very costly. In cases of insufficient datasets, the data augmentation method is applied. New data is being created from the existing one by applying operations like rotation, flipping, shifting, zooming, changing the brightness, and many more. This way the data is enlarged and the model performance is improved. Some of the augmentation techniques are:

- Flipping – the image can be flipped vertically or/and horizontally;
- Rotation – the image can be rotated according to the entered degrees;
- Zooming – the image can be zoomed in or/and zoomed out;
- Brightness – the image brightness can be changed to get darker or lighter.

An example of some augmentation techniques is shown in Fig. 3.1.

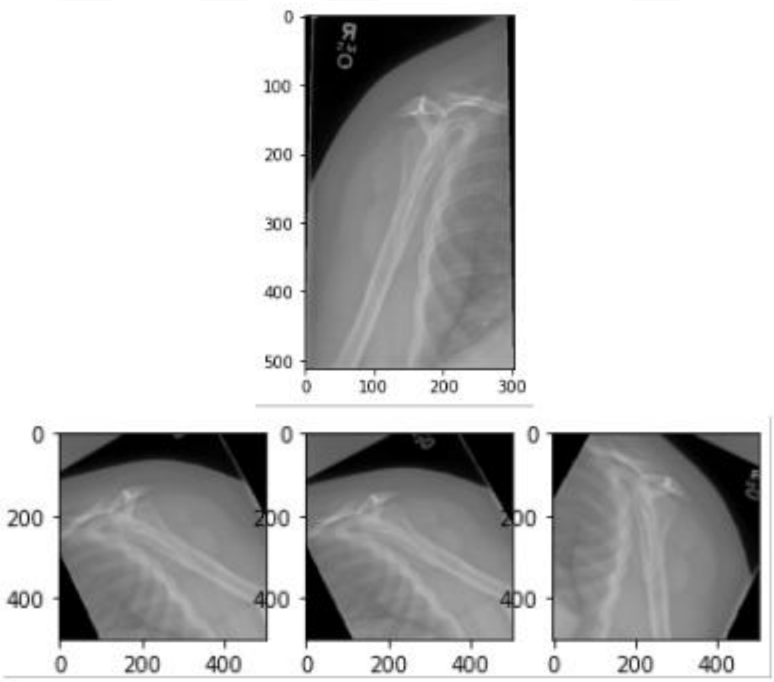


Figure 3.1 Example of image augmentation

### 3.2. Deep Learning

With the developing technology, terms such as artificial intelligence, machine learning, and deep learning are now daily used. We can say that the relationship between them is as this: Artificial intelligence (AI) is the science of making machines (computers) smart, machine learning is an approach to achieving artificial intelligence, and deep learning is a technique for applying machine learning as shown in Fig. 3.2.

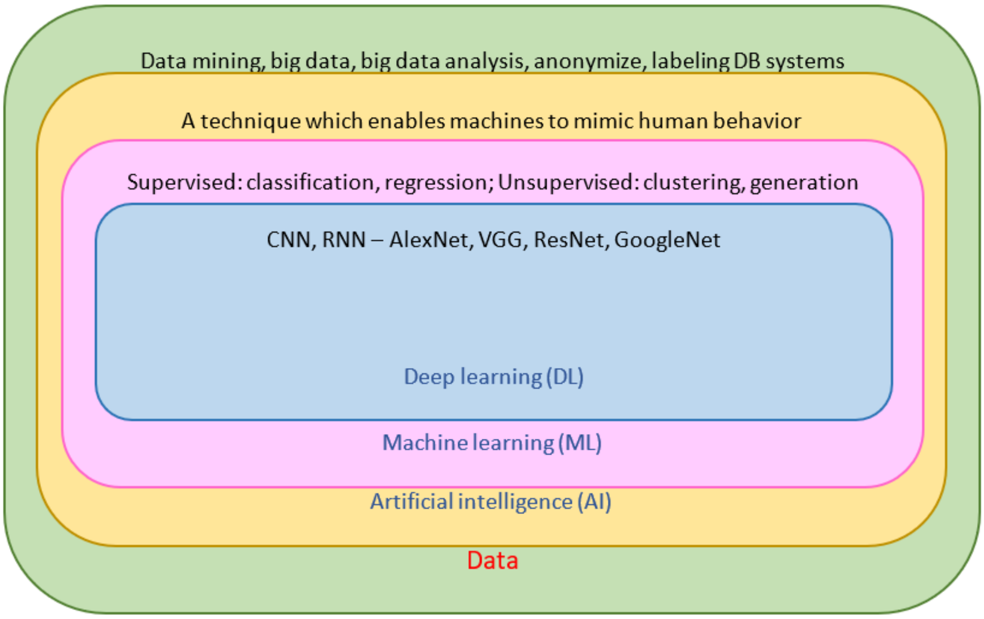


Figure 3.2. Medical image diagnosis method using artificial intelligence

DL is almost used in every field of life such as education, health, housing, work, and so on. It is capable to adapt to multiple data types, which makes it take place in classification, prediction, recognition of objects and images, applied in smart houses, autonomous vehicles, and so on. There are three categories of Deep Learning: supervised, unsupervised, and semi-supervised.

In supervised Deep learning, the model is trained with labeled datasets and the algorithm tries to produce the correct result from new input. It is used with classification methods to identify faces, and traffic symbols, make a prediction, classify images, or even convert speech to text.

In unsupervised learning, there is neither classified nor labeled data so it tries to learn inter-relations such as patterns and similarities to classify new data. The most used algorithm is clustering.

Semi-supervised learning is between the two methodologies. In this case, the training dataset contains labeled and unlabeled data. Some of the techniques used in deep learning and their application are shown in Fig. 3.3.

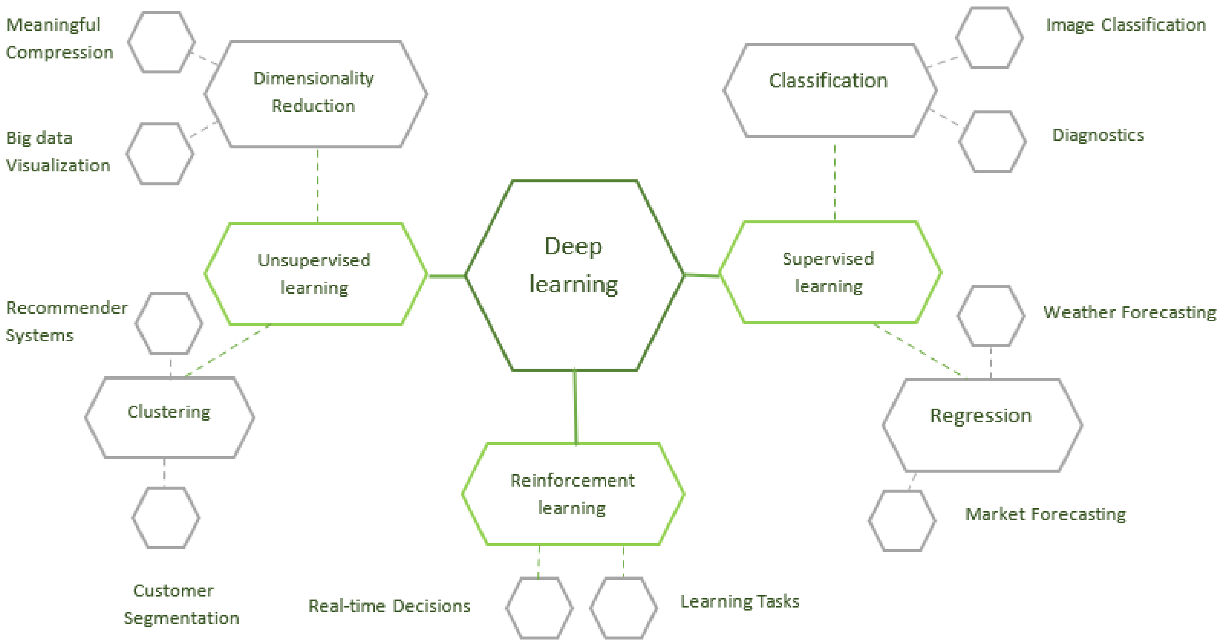


Figure 3.3. Some techniques used in deep learning and their application

### 3.2.1. Convolutional Neural Networks

Convolutional Neural Network, known as CNN, is mostly used in image processing and their analyses due to automated processing. CNNs are feedforward networks that are designed to work similarly to the human brain. Their workflow is only in one direction, from the input to the output. Normally the architecture is structured using modules of grouped convolutional and pooling layers, and a fully connected layer that can be more than one. An example of image classification using the CNN architecture is illustrated in Fig.3.4. The network takes an image as an input, after that several stages of convolution and pooling follow. As a final step, features are received from these layers by the FC (Fully connected) layers and it outputs the class label.

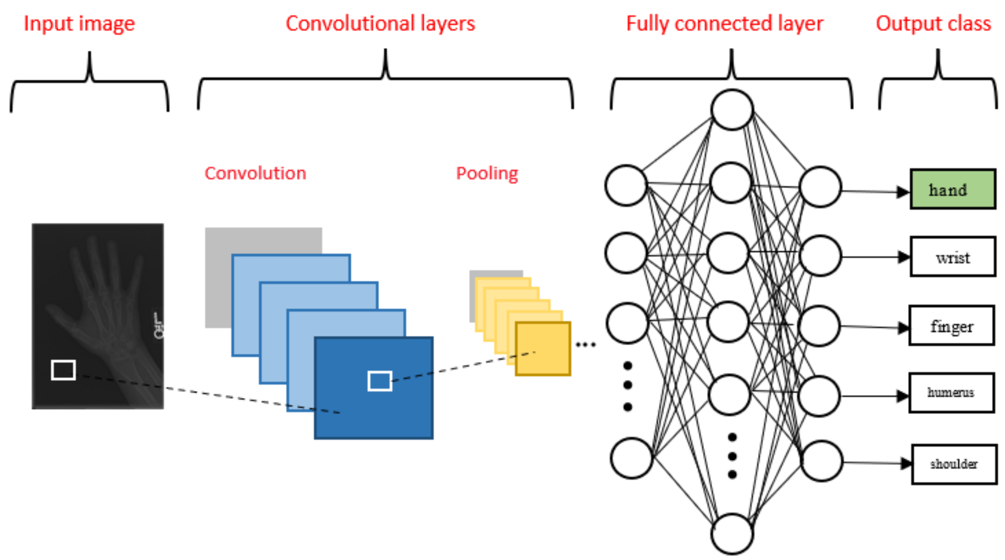


Figure 3.4 CNN image classification pipeline

**CNN layers**

In this section, the input layer, convolution layer, pooling layer, activation layer, and fully connected layers of CNN will be explained.

3.2.1.1 Input layer

The first layer in a CNN is the input layer. The images which are fed to the network should be represented as numerical data rather than pixels so the computer can understand them. If we have a gray-scale image with a height and width of 224x224 then its matrix size will be 224x224x1. In the case of colored images, the matrix will have a size of 224x224x3 having three channels for red, green, and blue. Each value in the matrix is between 0-255.

The size of the input image is important and reflects the output of the network. If we have large-sized images, this will increase the number of parameters that may cause network slowdown as it will use a lot of memory but it may give better results as more features can be extracted (Inik et al., 2017)

3.2.1.2. Convolutional layer

The most important component in CNN architecture is the convolutional layer. The main purpose of a convolutional layer is to detect features such as lines, edges, and other elements. A feature map is generated from this layer after applying convolutional filters (kernels) which can be of sizes 2×2, 3×3, 5×5, and so on. The filter is then like a square object that slides over the image. Using the weights of the filter the values are summed together as described in Fig.3.5 (Lee et al., 2017). The sliding size of the kernel is called a stride. When stride is set to 1, the filter is moved one by one; when set to 2, the filter skips two steps while sliding. Multiple filters are applied and so multiple feature maps are generated.

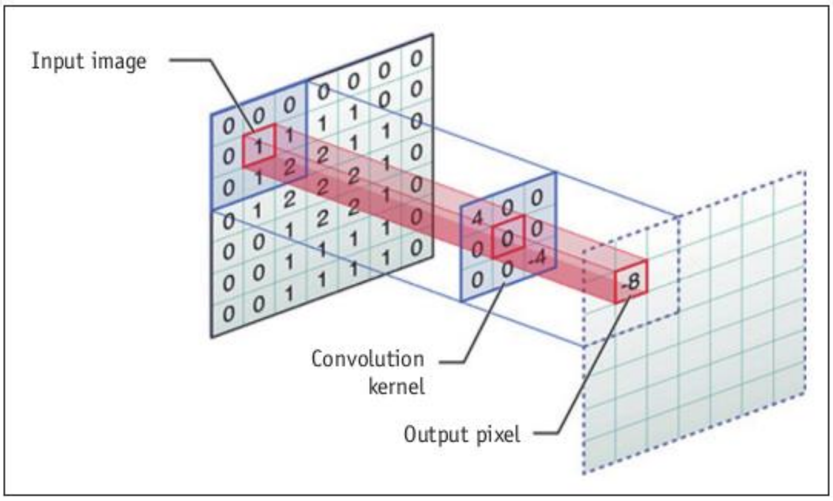


Figure 3.5 Illustration of convolution method (Lee et al., 2017)

Sometimes pixels with a value of zero are added as a border to the input images so the original input size is preserved. This technique is called Zero padding. As can be seen in Figure 3.6, the zero-padding affects the size of the matrix obtained as a result of the convolution operation. In the figure, the kernel is of size 3 with one stride and one padding.

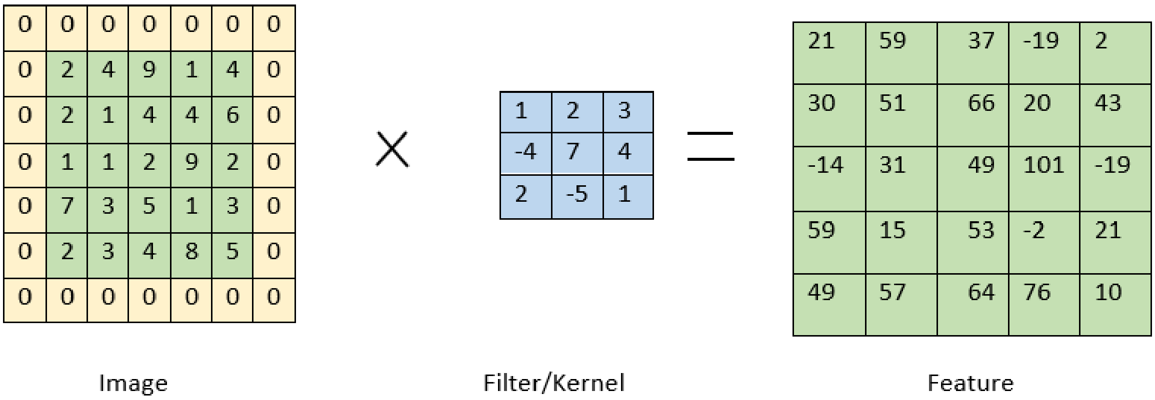


Figure 3.6 Convolution operation

### 3.2.1.3. Activation layer

Activation functions are used in a model to introduce non-linearity. This allows the model to learn new and more complex functional mappings between the inputs and variables. A feature map is formed by an activation function that processes the output of the convolution operation. Some of the most used non-linear operations are described below:

*Sigmoid*: It takes real numbers as input and produces an output between “0” and “1”. The sigmoid function can be represented mathematically by Eq. 3.2 and the graphic is shown in Fig. 3.7 (Naes et al., 1993).

$$f(x) = 1 / (1+e^{-x}) \tag{3.2}$$

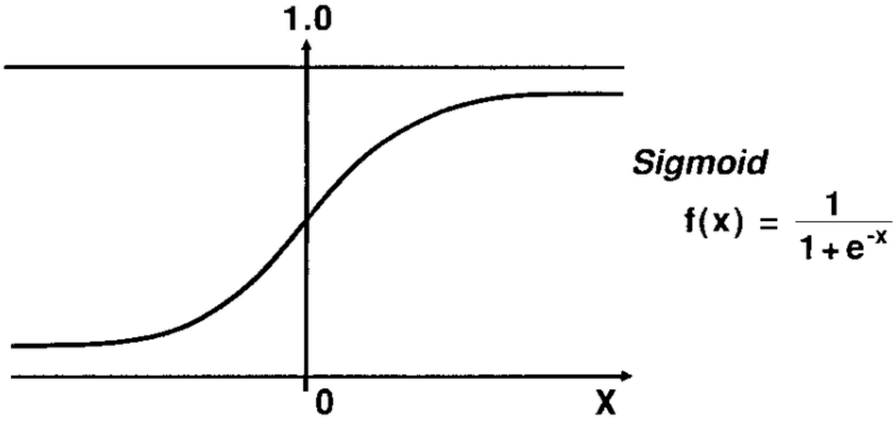


Figure 3.7 Sigmoid activation function (Naes et al., 1993)

*Tanh*: it takes real numbers as input, but the output is between − 1 and 1 as shown in Fig. 3.8 (Zayegh et al., 2018). The mathematical representation is given in Eq. 3.3.

$$f(x) \tanh = (e^{x} - e^{-x}) / (e^{x} + e^{-x}) \quad (3.3)$$

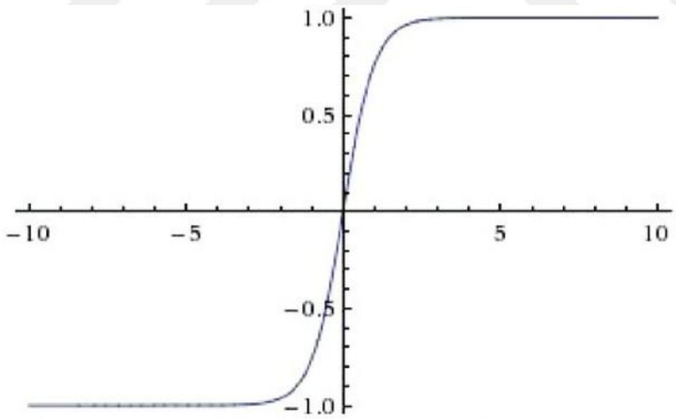


Figure 3.8 Tanh Activation Function (Zayegh et al., 2018)

*ReLU*: The Rectified Linear Unit is the most popular one used in CNN architectures. It takes the values of the input and converts them to positive numbers as shown in Fig.3.9 (Sultan et al., 2019). The mathematical representation is given below (Eq. 3.4) (Alzubaidi et al., 2021).

$$f\ (s)\ \text{ReLU} = \max(0, s) \tag{3.4}$$

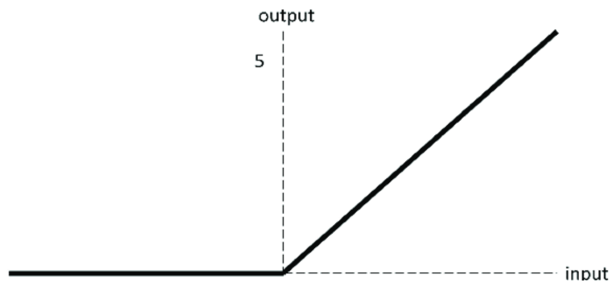


Figure 3.9 ReLU activation function (Sultan et al., 2019)

3.2.1.4. Pooling layer

The pooling layer is significantly used to reduce the size of the representation. This way the amount of computation and weights is decreased sufficiently. After applying the pooling layer, the size (height and width) of the input data is decreased but the number of channels does not change. Tree pooling, gated pooling, min pooling, max pooling, average pooling, global average, and global max-pooling are some of the pooling methods used (Alzubaidi et al., 2021). However, max pooling, average pooling, and global average pooling (GAP) are mostly preferred. They are shown in Fig.3.10.

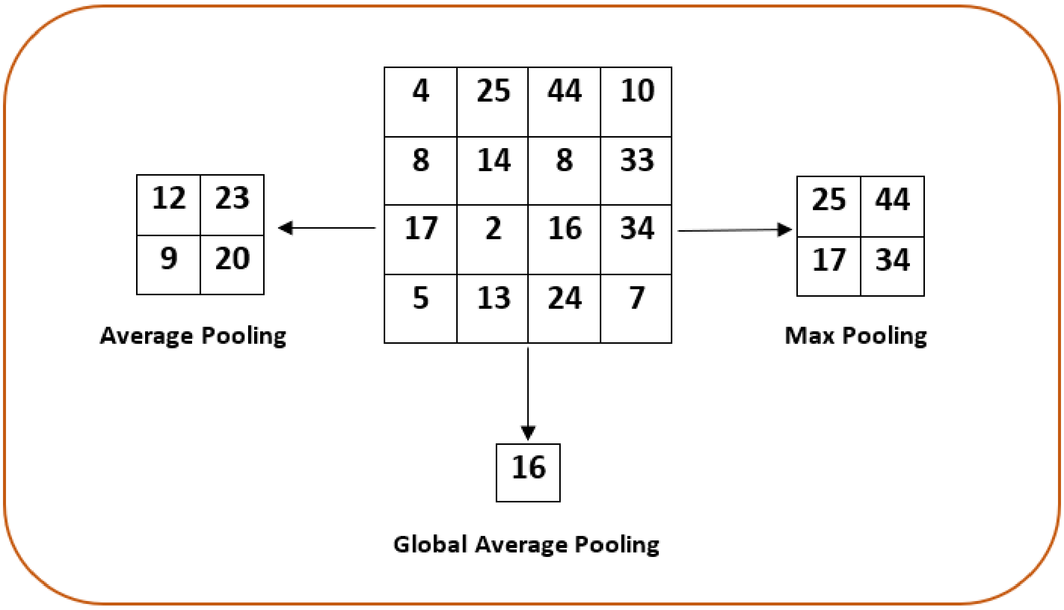


Figure 3.10 Three types of pooling operations

3.2.1.5. Fully connected layer

The Fully Connected (FC) layer is normally added at the end of the CNN architecture. The neurons in this layer are connected with all neurons in the preceding layer. The Fully Connected layer actually works as a classifier. A vector is created after flattening the feature maps and given as an input to the FC layer to produce an output that plays the role of the final output of CNN as shown in Fig. 3.11. The output size is chosen based on the application. For example, if the problem is binary, the output size will be 2, if the problem has 10 classes, the output will have 10 nodes, and so on.

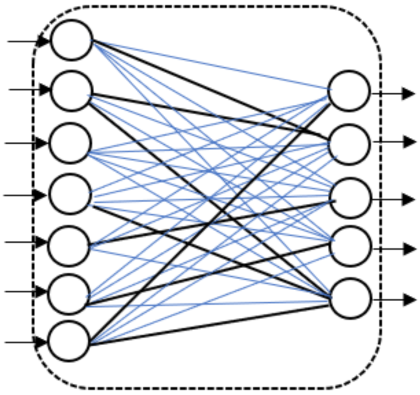

Figure 3.11 Visualization of FC layer

### 3.2.2. Transfer Learning

A big volume of data is essential when referring to a good performance of deep CNN models. Unfortunately, data is often limited due to the rareness, expensiveness to collect and label, or data inaccessibility. Here comes the need for transfer learning (TL). As its name indicates, previously gained information (knowledge) is transferred from one domain to another. The idea is to use CNN models trained on large datasets such as ImageNet, which contains more than one million images categorized in 1000 object classes (Russakovsky et al., 2015). Using TL is a time and power-saving technique due to the usage of the weights from the pre-trained models to initialize the weights of the new model as shown in Figure 3.12.

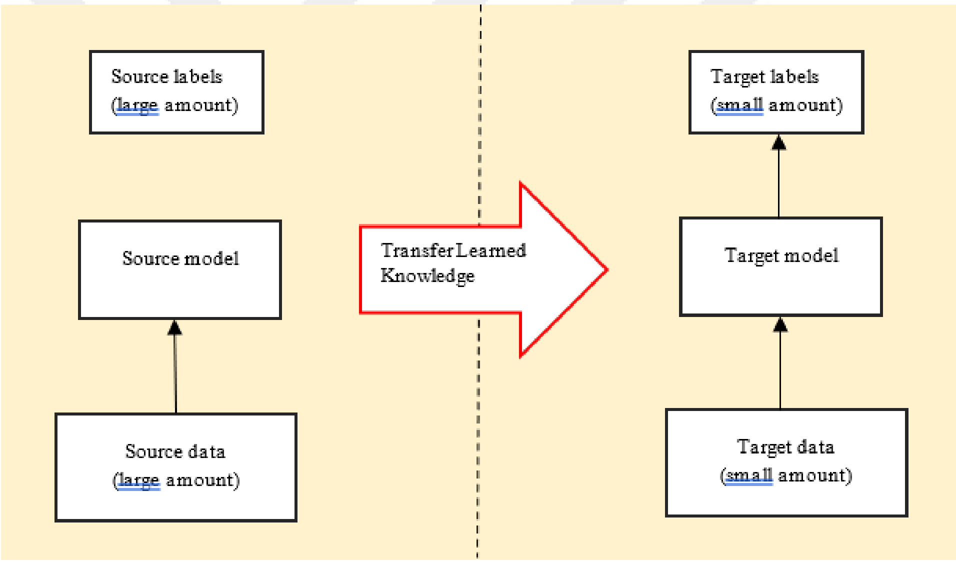


Figure 3.12 The idea of transfer learning

Transfer learning has been used prosperously in many different applications such as text sentiment classification (Wang et al., 2011), human activity classification (Harel et al., 2011), image classification (Zhu et al., 2011), multi-language text classification (Zhou et al., 2014), software defect classification (Nam et al., 2015), and others (Weiss et al., 2016).

**CNN architectures**

In this section, CNN models that have been used in this study will be discussed.

3.2.2.1. DenseNet

The authors of DenseNet proposed a model that maximizes the flow of information between the layers by connecting each layer to all layers in the network using a feed-forward approach. That means the network will have L.(L+1)/2 connections rather than L, as in the other architectures (Huang et al., 2018). This way feature propagation was strengthened but the model become parametrically high-priced due to the increased number of feature maps. Figure 3.13 shows the architecture of DenseNet (Lee et al., 2017).

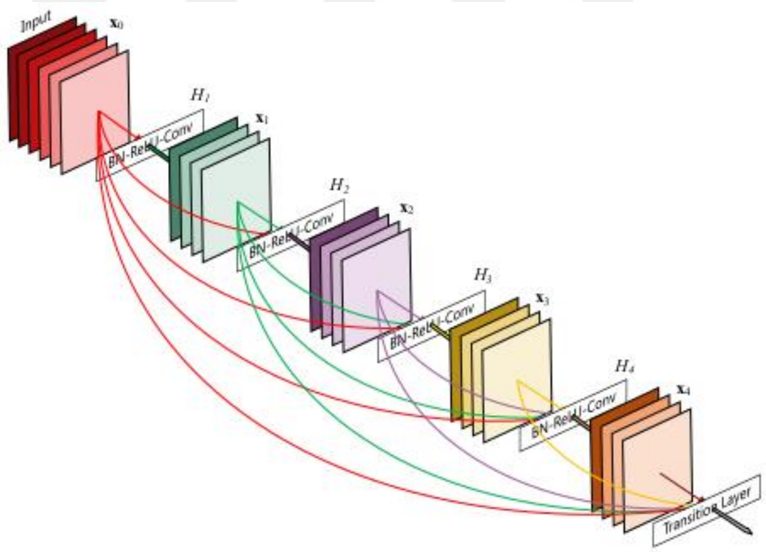


Figure 3.13 Layout of DenseNet (Lee et al., 2017)

The DenseNet architecture consists of convolutional layers, pooling layers, dense blocks, and a fully connected layer. Each dense block consists of two convolution layers: a 1x1 layer to reduce the feature maps and a 3x3 layer for improving computational efficiency. The layers between blocks are transition layers, which do convolution and pooling to achieve down-sampling of the network as shown in Fig. 3.14.

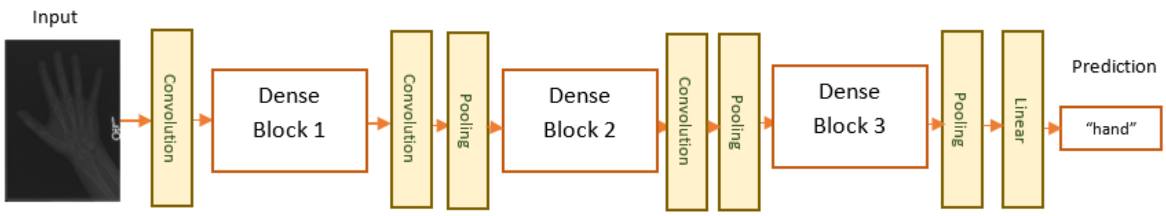


Figure 3.14 A DenseNet network containing three dense blocks.

The DenseNet model has different versions such as DenseNet-121, DenseNet-169, DenseNet-201, and DenseNet-224.

### 3.2.2.2. MobileNet

MobileNet is a model specifically designed for mobile and resource-constrained environments. What is special about this model is that it uses residual bottleneck convolution layers as basic blocks, after which the ReLU activation function is applied. The network uses a kernel of size 3 × 3 as it utilizes dropout and batch normalization during training (Sandler et al., 2019). The architecture of MobileNet is shown in Figure 3.15 (Paranjape et al., 2022).

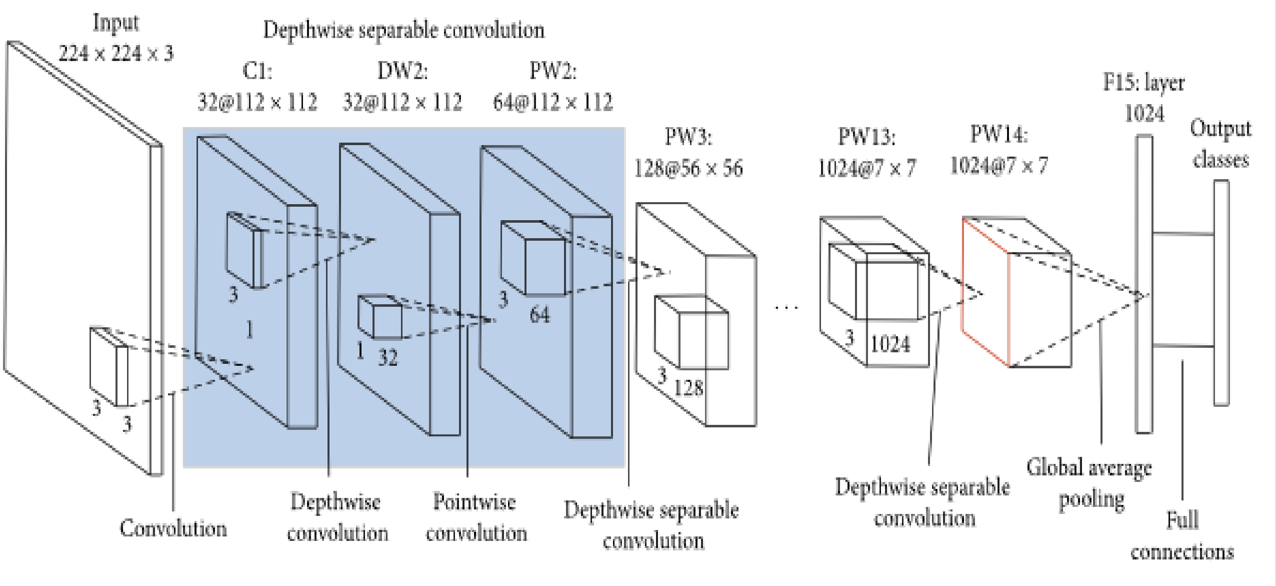


Figure 3.15 MobileNet architecture (Paranjape et al., 2022)

3.2.2.3. Xception

Xception architecture consists entirely of depthwise separable convolution layers with residual connections. It has 36 convolutional layers, structured into 14 modules, forming the feature extraction base (Chollet, F.,2017). The modules are divided into 3 components: entry, middle, and exit flow as shown in Fig. 3.16 (Harini et al., 2020). The network takes as input images of size 299x299x3. The depthwise separable convolution layers used in the Xception model are utilized for reducing the cost of convolution operation (Lo et al., 2019).

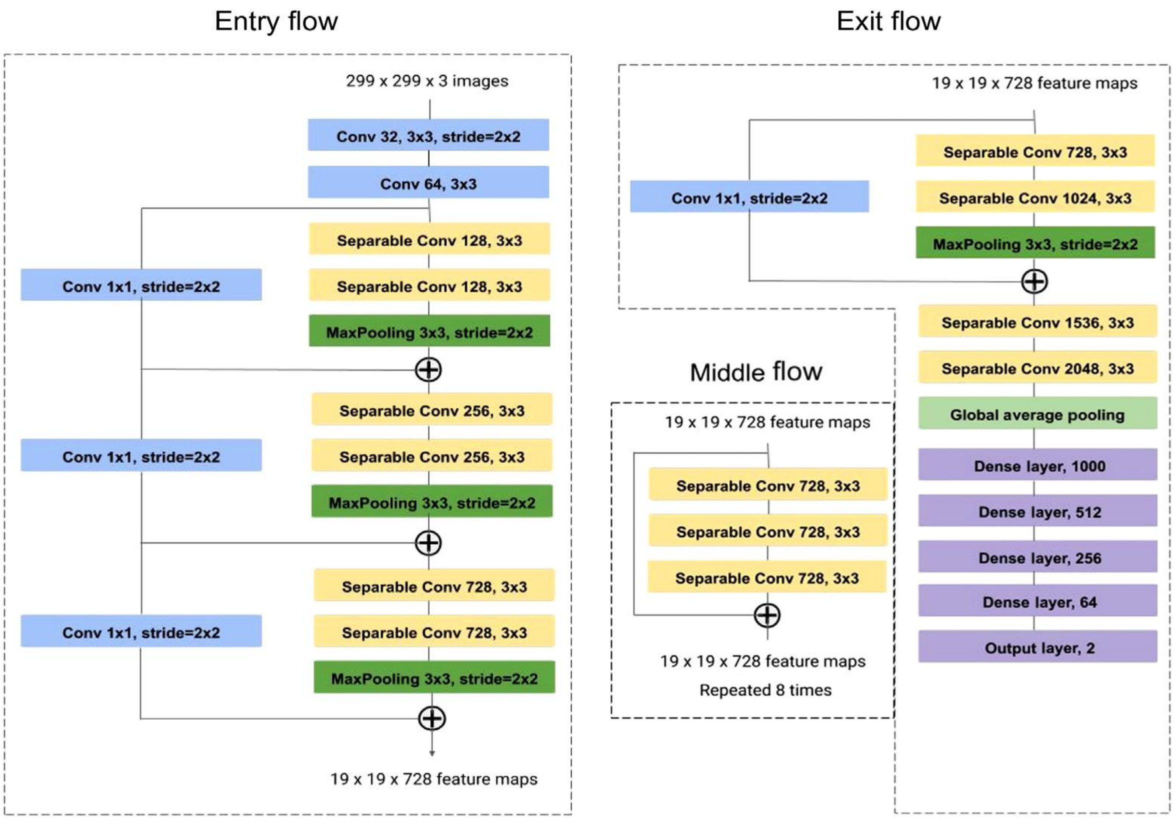


Figure 3.16 Xception model flow diagram (Harini et al., 2020)

3.2.2.4. EfficientNet

EfficientNet is the latest model presented in a research paper which focuses either on improving the accuracy and the efficiency of models. Trained on the ImageNet dataset

EfficientNet has significantly outperformed other ConvNets while being smaller and faster as illustrated in Figure 3.17 (Tan et al., 2020).

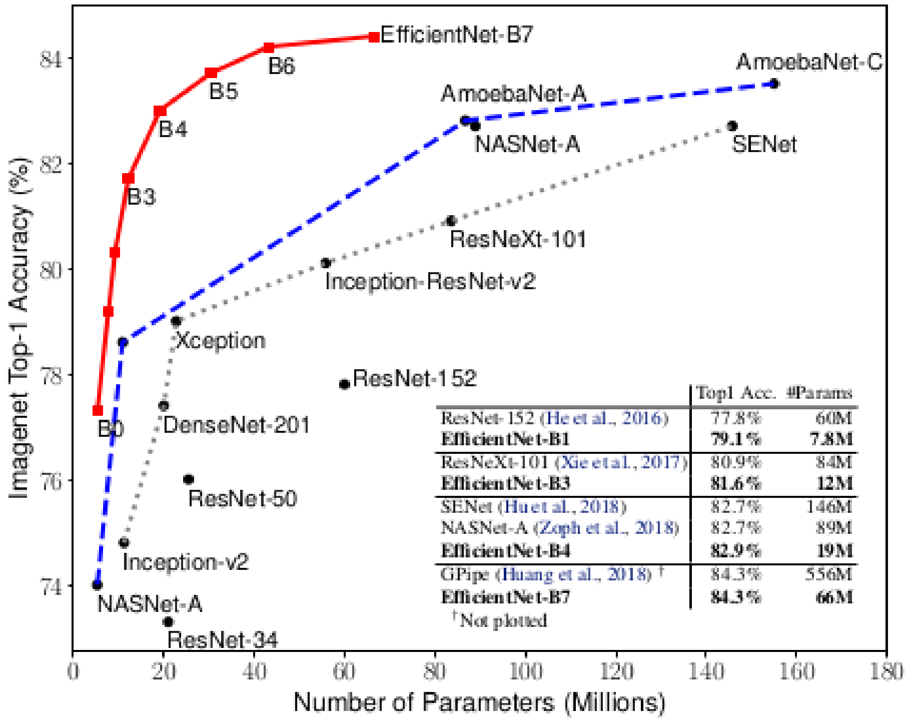


Figure 3.17 EfficientNet Model Size vs. ImageNet Accuracy (Tan et al., 2020)

Using coefficients that are fixed, a method scales the width, depth, and resolution of the network uniformly (Tan et al., 2020). Depth is equivalent to the number of layers in the network, Width is simply the number of channels in a convolutional layer and Resolution is the resolution of the input image. In Figure 3.18 the scaling method is shown compared to conventional methods (Tan et al., 2020).

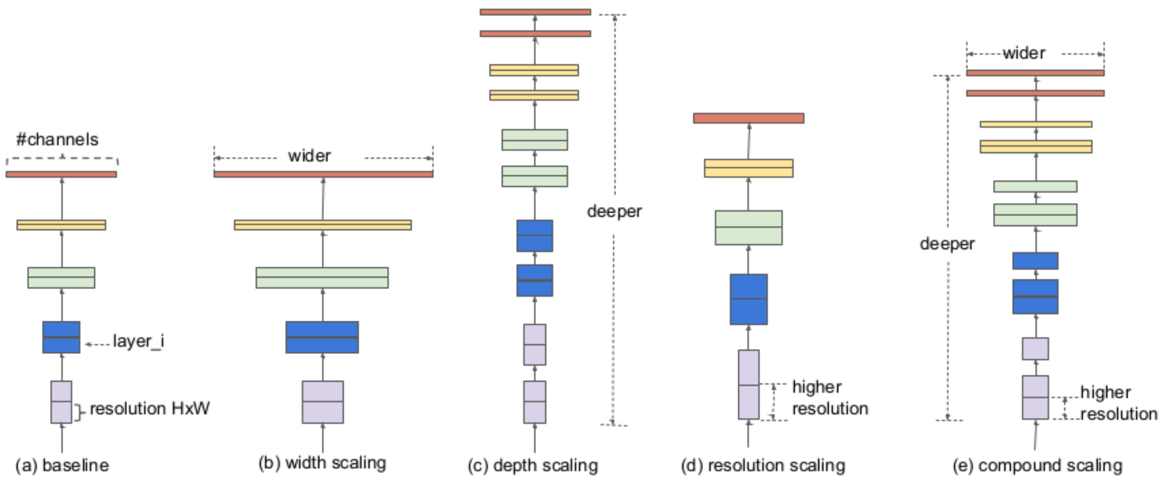


Figure 3.18 Model Scaling in EfficientNet. (a) is the baseline; (b)-(d) are only width, depth, or resolution scaling; (e) is the proposed scaling method in EfficientNet (Tan et al., 2020)

### 3.2.2.5. ResNet

The Residual Neural Network (ResNet) was first introduced by Kaiming He et al. in their research paper "Deep Residual Learning for Image Recognition". The architecture of ResNet was inspired by the VGG network, with the convolutional networks having 3×3 filters. However, the difference is that the ResNet has lower complexity and fewer layers compared to the VGG network. The authors inserted shortcut connections, that gave the network its residual version. The idea here is when the input has the same dimension as the output, identity shortcuts are used. When the dimensions increase, two options were being considered. The first option was that extra zero entries would be padded for increasing dimensions and the shortcut would perform identity mapping (He et al., 2015). In the second option, the projection shortcut was used to match dimensions.

In the architecture of ResNet50, which is used in this thesis, the 2-layer blocks were replaced by 3-layer bottleneck blocks. In this way, the architecture of Resnet50 was formed, resulting in higher accuracy and lesser training time.

### 3.2.3. Fine-tuning: Early stopping

Fine-tuning is used to learn the new features and work with the newly added data on top of the pre-trained dataset (Figure 3.19). Optionally the layers of the model or some of them could be unfrozen and the model is trained again, but this time, with a low learning rate to avoid overfitting. When the validation accuracy is no longer getting improved, a callback technique Early Stopping is used to stop the training (Chauhan et al., 2021). In most cases, fine-tuning gives better accuracy results and the model performance improves noticeably.

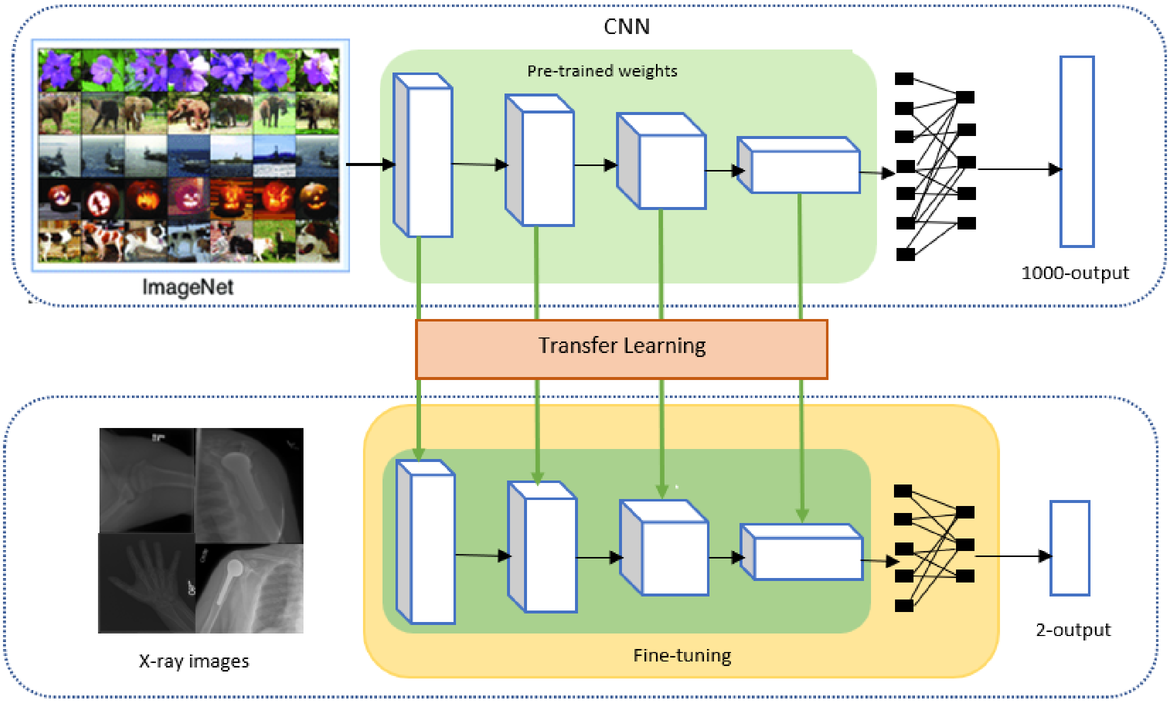


Figure 3.19 Fine-tuning

### 3.2.4. Hyperband Hyperparameter optimization

For every application, researchers have to choose a combination of hyperparameters that will be set to the model, so it gives the best results. This process of trying different combinations and choosing the best set of hyperparameters to achieve the best model performance is called Hyperparameter Optimization. In this work, Hyperband Hyperparameter Optimization is used and explained in the next section.

The creators of the method Hyperband Hyperparameter Optimization state that their main focus was on speeding up random search through adaptive resource allocation and early stopping (Li et al., 2018). When using the Hyperband method, a resource must be selected in advance, be it iterations, data samples, or characteristics, which is allocated to randomly selected configurations. The model begins training with the configurations, removing the badly performed ones.

Hyperband attempts to solve the issue with successive halving (Li et al., 2018). The main idea behind successive halving is that a budget B is allocated to a set of hyperparameter configurations and then only the best half of them is being kept and the other half is thrown out. This process is repeated until only one model is left.

### 3.3. Evaluation Metrics

Evaluation metrics are used to measure how efficient is the created classifier. To evaluate the proposed model and compare it with other works, first, we need to explain some basic terms:

- TP (True positive): represents the rightly classified images as normal;
- TN (True negative): represents the rightly classified images as abnormal;
- FP (False positive): represents the wrongly classified images as normal;
- FN (False negative): represents the wrongly classified images as abnormal.

All metrics used to evaluate performance are provided below.

Accuracy: the ratio of the rightly classified images to all the images as the following definition shows (Eq. 2.5):

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN) \quad (2.5)$$

Precision (P): the ratio of correct positive predictions to all positive predictions (Eq. 2.6).

$$\text{Precision} = TP / (TP + FP) \quad (2.6)$$

Recall (R): calculates the fraction of positive patterns that are correctly classified (Eq. 2.7).

$$\text{Recall} = TP / (TP + FN) \tag{2.7}$$

F1-score: Denotes the harmonic average between recall and precision rates (Eq. 2.8).

$$\text{F1-score} = 2 \times (P \times R) / (P + R) \tag{2.8}$$

Cohen's kappa statistic (κ): is used to measure the level of agreement between two raters (Eq. 2.9).

$$(p_0 - p_e)/(1 - p_e) \tag{2.9}$$

where $p_0$ is the result of the accuracy, and $p_e$ is the probability of random agreement.

Confusion matrix: it represents the performance of a classification algorithm, consisting of the four characteristics TP, TN, FP, and FN, as shown below in Table 3.1:

Table 3.1 Confusion matrix visualization

| | Predicted 0 | Predicted 1 |
|---|---|---|
| Actual 0 | **TN** | **FP** |
| Actual 1 | **FN** | **TP** |

AUROC (Area Under the Receiver Operating Characteristics): represents the capability of the model to distinguish between classes. The higher values of AUROC the better the model is at distinguishing between the classes (Fig. 3.20). The ROC curve shows the rate between the true positives and the true negatives.

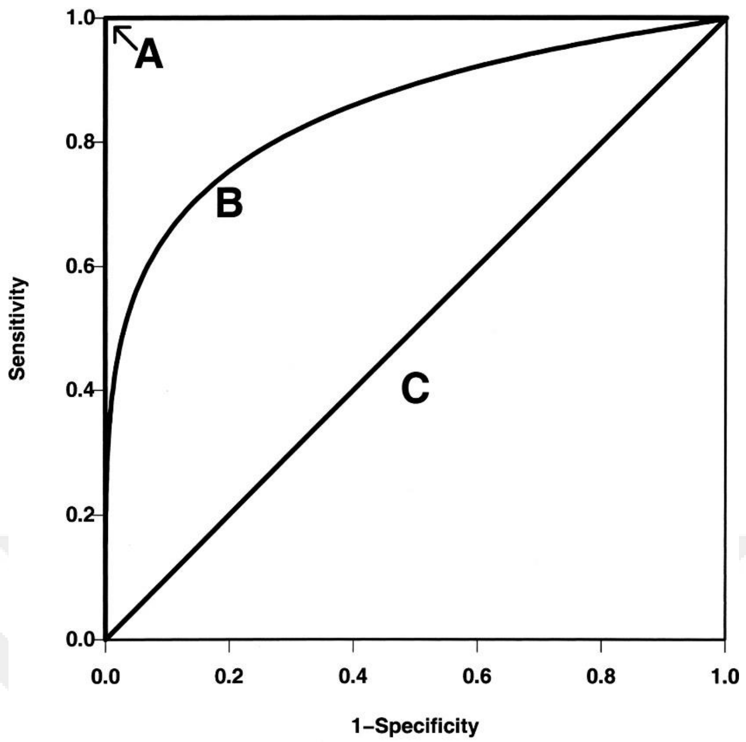


Figure 3.20 AUROC visualization: A referring to the gold standart, B - a typical ROC curve, and C corresponding to random chance (Zou et al., 2007)

# 4. DATASET

The dataset used in this study is the MURA dataset published in 2018 by Stanford ML Group. It is a large dataset of bone X-rays containing 14,656 musculoskeletal studies of the upper extremity including the shoulder, humerus, elbow, forearm, wrist, hand, and finger (Rajpurkar et al., 2018). The images belong to 12,225 patients. All the images are labeled as normal or abnormal manually by radiologists from “Stanford Hospital” between 2001 and 2012.

As it was used in a competition, the test images are not given public. Therefore, there are only two folders, namely Train and Validation, in this dataset. The training subset contains 36,808 images, from which 21,935 images are labeled as normal and 14,873 as abnormal. To achieve results close to the ones gained in the competition, 320 images are used for testing. Hence, from a total of 3,197 images in the validation folder, 2,877 images are used for validation and the other 320 for testing. The Validation data consists of 1497 normal and 1380 abnormal images and the test data has 170 normal and 150 abnormal images as shown in Table 4.1. The dataset in this form was used in the study as dataset-1

Table 4.1 Distribution of MURA dataset

| type | train | validation | test |
|---|---|---|---|
| normal | 21 935 | 1497 | 170 |
| abnormal | 14 873 | 1380 | 150 |
| total | 36 808 | 2877 | 320 |

Figure 4.1 and Figure 4.2 show samples from the seven categories labeled as normal, and abnormal respectively. Table 4.2 summarizes the distribution of images according to the upper extremity they belong to.

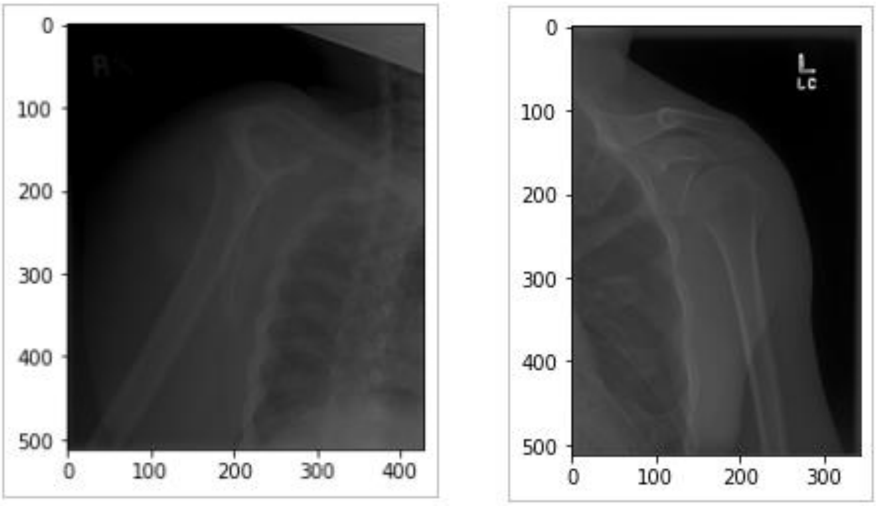

Figure 4.1 Normal labeled X-ray images

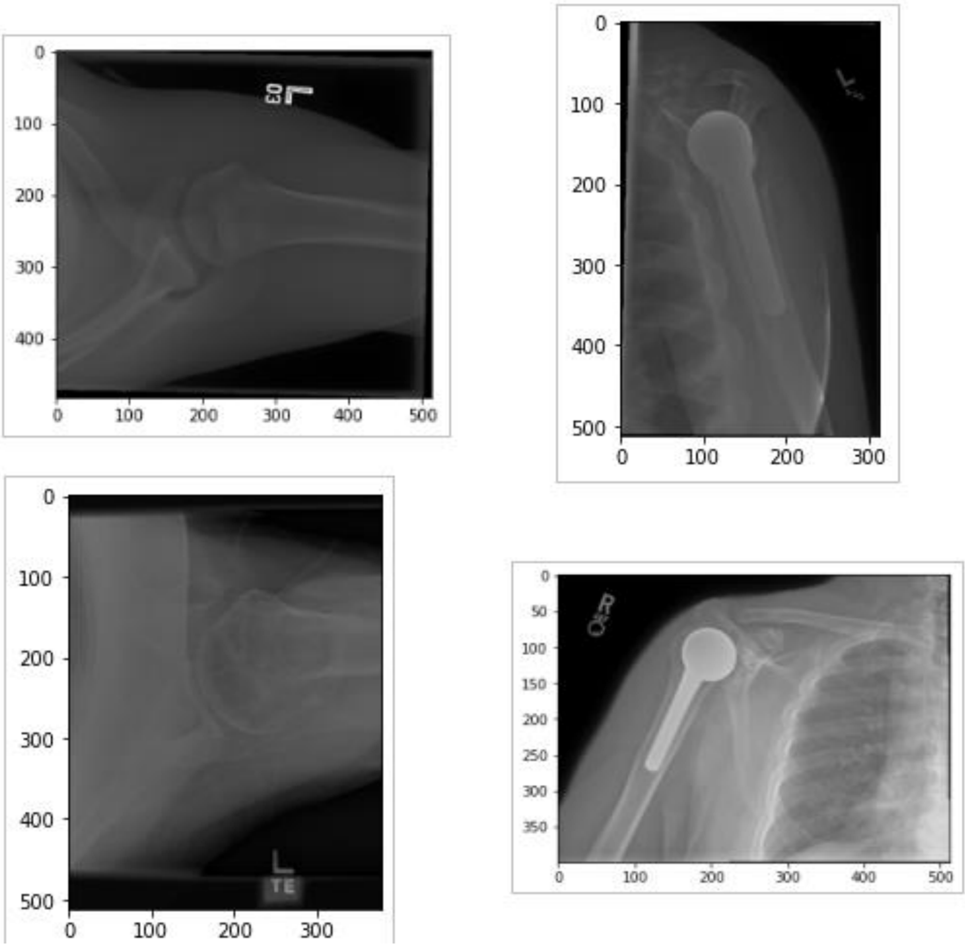

Figure 4.2 Abnormal labeled X-ray images

Table 4.2 Distribution of images according to the upper extremity

| Category | train | validation | test |
|---|---|---|---|
| Elbow | 4931 | 419 | 46 |
| Finger | 5106 | 412 | 49 |
| Hand | 5543 | 419 | 41 |
| Humerus | 1272 | 260 | 28 |
| Forearm | 1825 | 267 | 34 |
| Shoulder | 8379 | 498 | 65 |
| Wrist | 9752 | 602 | 57 |

It can be seen from the plot in Fig.4.3 that the data is imbalanced in this dataset. The number of normal cases exceeds the number of abnormal cases: the negative images represent 59% of the dataset, and the positive images represent 41%. For a balanced evaluation, the dataset was divided into three parts, including training, test, and validation. By using a common rule of splitting 0.6:0.2:0.2, 24003, 8001, and 8001 samples for train, test, and validation sets, were obtained respectively. The dataset in this form is used in the study as dataset -2.

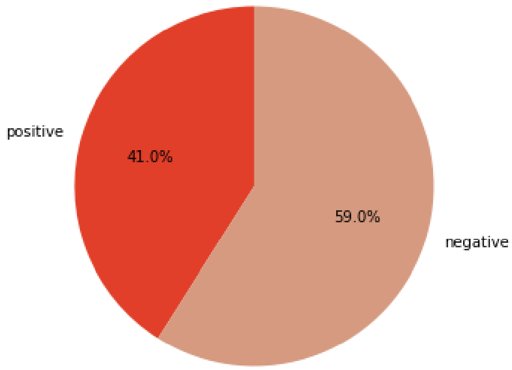


Figure 4.3 Imbalanced data illustration

# 5. PROPOSED MODELS

X-ray image processing techniques are widely used to mainly detect and classify anomalies of the bones and chest. There are three deep learning-based methods for classification: constructing and training a new network from scratch, using transfer learning with pre-trained CNNs, or using classic classifiers such as SVM, Random forest (RF), and Decision tree (DT) with features extracted by CNN.

In this thesis, using the Transfer learning methodology, two different approaches are proposed for the classification of the upper extremity musculoskeletal X-ray images. The first one is utilizing five different networks: DenseNet-169, DenseNet-201, Xception, MobileNet, and EfficientNet, to classify the images. The second approach is utilizing an ensemble model of three different networks: DenseNet-201, EfficientNetB4, and EfficientNetV2M.

The deep learning models in this study are implemented in the Keras library, designed for Python's deep learning applications. The reason why Keras is preferred is that it is user-friendly and easy to use in deep learning studies. Since the size of the dataset containing the musculoskeletal X-ray images used in this study is quite large and speed is an important concept in deep learning studies, it is preferred that the machine on which the studies are carried out becomes a powerful one. However, since we do not have such an infrastructure for this study, all studies were carried out on a cloud platform called Google Colab.

## 5.1. Hyperband Optimization Algorithm

Before starting to train the models, we need to choose a combination of hyperparameters. To have the best combination set, a hyperparameter optimization algorithm, called Hyperband, was utilized in this thesis. The Hyperband optimization algorithm was used to optimize three of the parameters of the model: activation function, learning rate, and batch size. It had two choices for the activation function – softmax and sigmoid, the learning rate had the values 0.01, 0.001, or 0.0001, and the batch size had to be one of 16 or 32. Each of the pre-trained CNNs was trained with a maximum of 40 epochs to get the optimal hyperparameters. The results are summarized in Table 5.1. The optimal learning rate for the optimizer for all the models was 0.001. All the networks got “16” as an optimal batch size except for the Xception network with a 32 bach size. DenseNet169, Xception and MobileNetV2 had an optimal activation function “softmax”, whereas DenseNet201 and EfficientNetB5 had “sigmoid” as an activation. These results were being used while training the networks with Transfer learning.

Table 5.1 The results achieved by applying Hyperband optimization algorithm

| **CNN** | **Activation** | **Learning rate** | **Batch size** |
|---|---|---|---|
| DenseNet169 | softmax | 0.001 | 16 |
| DenseNet201 | sigmoid | 0.001 | 16 |
| Xception | softmax | 0.001 | 32 |
| MobileNetV2 | softmax | 0.001 | 16 |
| EfficientNetB5 | sigmoid | 0.001 | 16 |

### 5.2. Classification with Transfer Learning

In the first experiment, the feature extraction method was used. Each of the five models, DenseNet-169, DenseNet-201, Xception, MobileNet, and EfficientNet, used in this study were already pretrained using the ImageNet dataset. Firstly, the fully connected layer was removed and a new one was added with an output equal to the number of classes in the dataset. The pre-trained layers of all the models were frozen and only the last added layers were being trained. The variable-sized X-ray images were resized since all pre-trained networks require an input size of 224x224x3. Before feeding the images into the network, they were normalized and standardized to have the same mean and standard deviation as images in the ImageNet dataset. Horizontal flip, zooming, and rotation to 30 degrees were applied for data augmentation. As a final step, the networks were trained for 50 epochs using training and validation images so that images can be classified. All the models were trained using an Adam optimizer with a learning rate of 0.001. Finally, the evaluation metrics described in Section 2.3, were used to measure the performance of the networks on test images that were not used in the training phase.

### 5.2.1. Performance improvement tricks

In the second experiment, a trick to improve the learning capacity of the deep learning model was used. For this purpose, after the network was first frozen and trained with 50 epochs, the pre-trained layers of all the models were unfrozen and the whole model was retrained but with a smaller learning rate (lr=0.00001). This performance improvement trick is also known as Fine-tuning.

All five models, DenseNet-169, DenseNet-201, Xception, MobileNet, and EfficientNet, were set to be trained for 30 epochs. If the validation accuracy has not improved for 5 epochs the EarlyStopping callback was called and the training was stopped to avoid overfitting. The evaluation metrics described in Section 2.3, were used to measure the performance of the networks on test images that were not used in the training phase.

## 5.3. Image Preprocessing with CLAHE

The CLAHE method divides the image into 8x8 pieces and performs histogram equalization within each frame. Since the images in the dataset have RGB channels, the CLAHE method cannot be applied directly. Therefore, first of all, the color channels of the images need to be changed. They were converted to the LAB color map because it contains the Lightness channel which shows the contrast between black and white in an image. After that the CLAHE method was applied, and the channels of the images were again converted to the RGB color space. An X-ray sample image is shown in Figure 4.1 before and after applying the CLAHE method. After this step, the dataset is fed to the same five CNNs to apply Transfer learning. The evaluation metrics described in Section 2.3, were used to measure the performance of the networks on test images that were not used in the training phase.

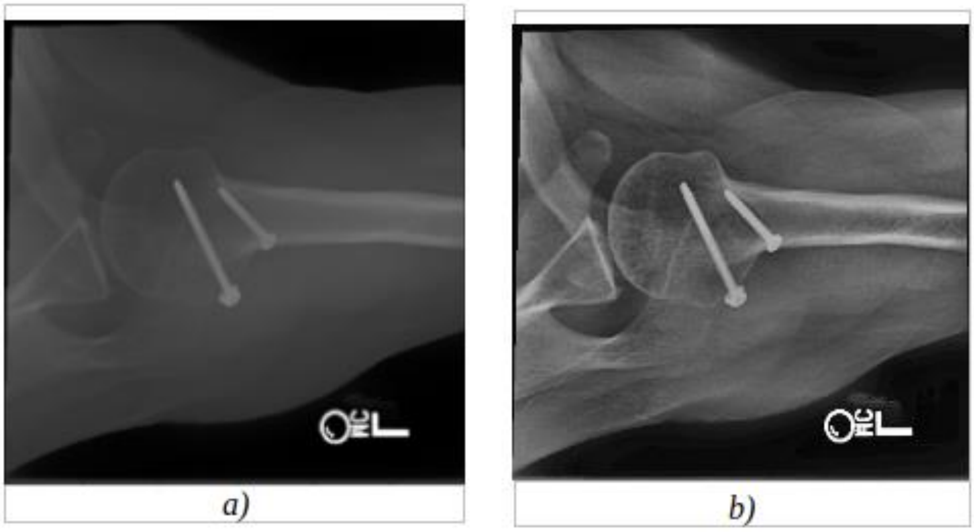


Figure 5.1 Image pre-processing a) original X-ray image b) X-ray image after applying CLAHE method

### 5.4. Ensemble Learning

In this section, the commutations and compromises of combining different CNN buildings, to improve anomaly detection in patients' X-Ray images, were evaluated. For a balanced evaluation, the dataset was divided into three parts, including training, test, and validation. By using a common rule of splitting 0.6:0.2:0.2, 24003, 8001, and 8001 samples for train, test, and validation sets, were obtained respectively. To explore the performance of the ensemble learner, the dataset was trained with five CNNs: DenseNet-201, Resnet50V2, MobileNetV2, EfficientNetB4, and EfficientNetV2M. Combining the results of these networks, an ensemble model was built.

# 6. RESULTS AND DISCUSSION

## 6.1. Classification Results with Transfer Learning

In this section, the training was performed with dataset-1. Training Accuracy, Training loss, and Test Accuracy are some of the parameters to evaluate the performance of a model. The training accuracies and training losses for the architectures DenseNet-169, DenseNet-201, Xception, MobileNet, and EfficientNet are given in Table 6.1. It can be seen from the table that the DenseNet-201 network gives the highest training accuracy. In addition to this, the DenseNet-201 network had the highest test accuracy of 77%. Also, the precision of the normal class, recall of the abnormal class, and F1 score of the abnormal class are the highest for the DenseNet-201 network. The test results are summarized in Table 6.2.

Table 6.1 Training accuracies and training losses of five models (Dataset-1)

| **Architectures** | **Training accuracy** | **Training loss** |
|---|---|---|
| DenseNet169 | 0.73 | 0.54 |
| DenseNet201 | 0.74 | 0.54 |
| Xception | 0.73 | 0.54 |
| MobileNetV2 | 0.70 | 0.58 |
| EfficientNetB5 | 0.73 | 0.53 |

Table 6.2. Comparison of performance metrics of five models on test data (Dataset-1)

| **Architectures** | **Overall accuracy** | **Class** | **Precision** | **Recall** | **F1-score** |
|---|---|---|---|---|---|
| DenseNet169 | 0.71 | Normal | 0.67 | 0.91 | 0.77 |
| | | Abnormal | 0.83 | 0.48 | 0.61 |
| DenseNet201 | **0.77** | Normal | 0.77 | 0.78 | 0.77 |
| | | Abnormal | 0.74 | 0.74 | 0.74 |
| Xception | 0.69 | Normal | 0.71 | 0.71 | 0.71 |
| | | Abnormal | 0.67 | 0.67 | 0.67 |
| MobileNetV2 | 0.76 | Normal | 0.76 | 0.80 | 0.78 |
| | | Abnormal | 0.76 | 0.71 | 0.74 |
| EfficientNetB5 | 0.72 | Normal | 0.67 | 0.91 | 0.77 |
| | | Abnormal | 0.83 | 0.49 | 0.62 |

According to Table 6.3, DenseNet-201 achieved the best results for Cohen’s kappa and AUROC metrics.

Table 6.3. Cohen’s kappa and AUROC results of five models (Dataset-1)

| **Architectures** | **Cohen’s kappa score** | **AUROC** |
|---|---|---|
| DenseNet169 | 0.40 | 0.69 |
| DenseNet201 | **0.55** | **0.77** |
| Xception | 0.38 | 0.69 |
| MobileNetV2 | 0.51 | 0.76 |
| EfficientNetB5 | 0.41 | 0.70 |

### 6.1.1. Classification results with the performance improvement trick

After applying the performance improvement trick discussed in Section 5.2.1, the overall results got better for all five networks. In this section, the training was performed with dataset-1. It can be seen from Table 6.4 that the DenseNet-201 network has again achieved the highest training accuracy of 93% and also the highest test accuracy of 86% as given in Table 6.5.

Table 6.4 Training accuracies and training losses of five models after applying Fine-tuning (Dataset-1)

| Architectures | Training accuracy | Training loss |
|---|---|---|
| DenseNet169 | 0.90 | 0.23 |
| DenseNet201 | 0.93 | 0.18 |
| Xception | 0.85 | 0.34 |
| MobileNetV2 | 0.84 | 0.36 |
| EfficientNetB5 | 0.91 | 0.21 |

The DenseNet-201 network had also good values for precision, recall, and F1-score for both classes compared to the other networks. The performance results of the five models with the MURA dataset after applying fine-tuning are shown in Table 6.5.

Table 6.5 Comparison of performance metrics of five models on test data after applying Fine-tuning (Dataset -1)

| Architectures | Overall accuracy | Class | Precision | Recall | F1-score |
|---|---|---|---|---|---|
| DenseNet169 | 0.77 | Normal | 0.72 | 0.92 | 0.81 |
| | | Abnormal | 0.87 | 0.59 | 0.70 |
| DenseNet201 | **0.86** | Normal | 0.77 | 0.94 | 0.84 |
| | | Abnormal | 0.91 | 0.67 | 0.77 |
| Xception | 0.82 | Normal | 0.77 | 0.95 | 0.85 |
| | | Abnormal | 0.92 | 0.69 | 0.79 |
| MobileNetV2 | 0.79 | Normal | 0.73 | 0.96 | 0.83 |
| | | Abnormal | 0.94 | 0.59 | 0.73 |
| EfficientNetB5 | 0.81 | Normal | 0.75 | 0.95 | 0.84 |
| | | Abnormal | 0.92 | 0.65 | 0.76 |

The enhancement can be seen also in Cohen's kappa score, which improved from 55% to 62%, and in AUROC with an improvemenet from 77% to 81%. Although DenseNet-201 has achieved the highest accuracy, it can be seen in Table 6.6 that Xception network had the best Cohen's kappa score and AUROC. However, the difference between the values achieved from the two networks is small. In the resulting analysis of the two models, the performance of the DenseNet-201 is comparatively better.

Table 6.6 Cohen's kappa and AUROC results of five models after applying Fine-tuning (Dataset-1)

| **Architectures** | **Cohen's kappa score** | **AUROC** |
|---|---|---|
| DenseNet169 | 0.52 | 0.75 |
| DenseNet201 | **0.62** | **0.81** |
| Xception | **0.64** | **0.82** |
| MobileNetV2 | 0.57 | 0.78 |
| EfficientNetB5 | 0.60 | 0.80 |

Training the models with fine-tuning has led to improved results. This can be clearly seen in Figure 6.1, which shows the accuracies obtained from the networks with Transfer learning and fine-tuning as an additional step. The improvement is obvious in all the networks with a minimum increase of 3% in MobileNetV2 and a maximum increase of 13% in the Xception model.

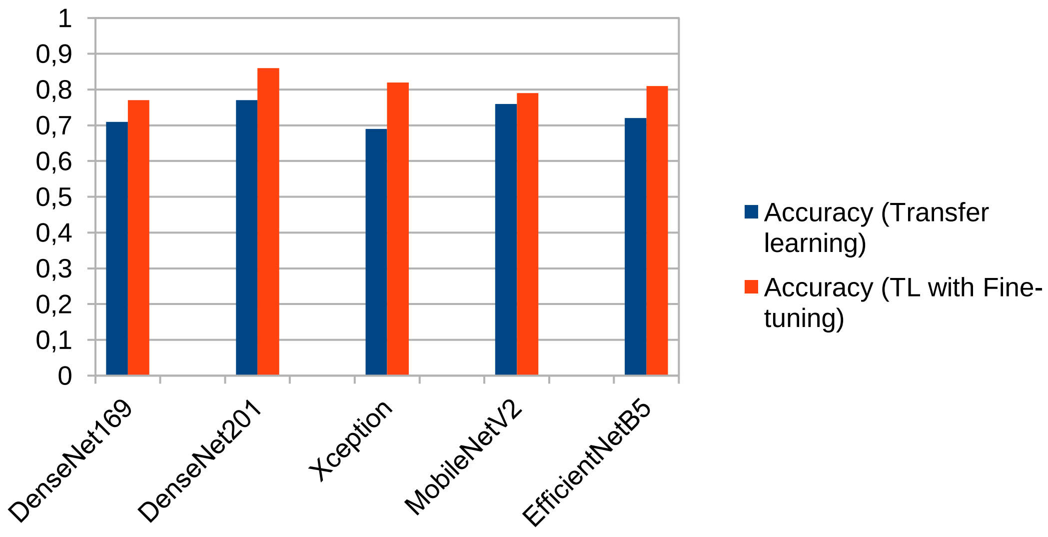


Figure 6.1. Comparison of the accuracies obtained from Transfer learning and Fine-tuning methods

The Cohen’s kappa score and the AUROC also show the improved performance of the models after training them with Fine-tuning. The comparison is illustrated in Figure 6.2 and Figure 6.3 for the two metrics.

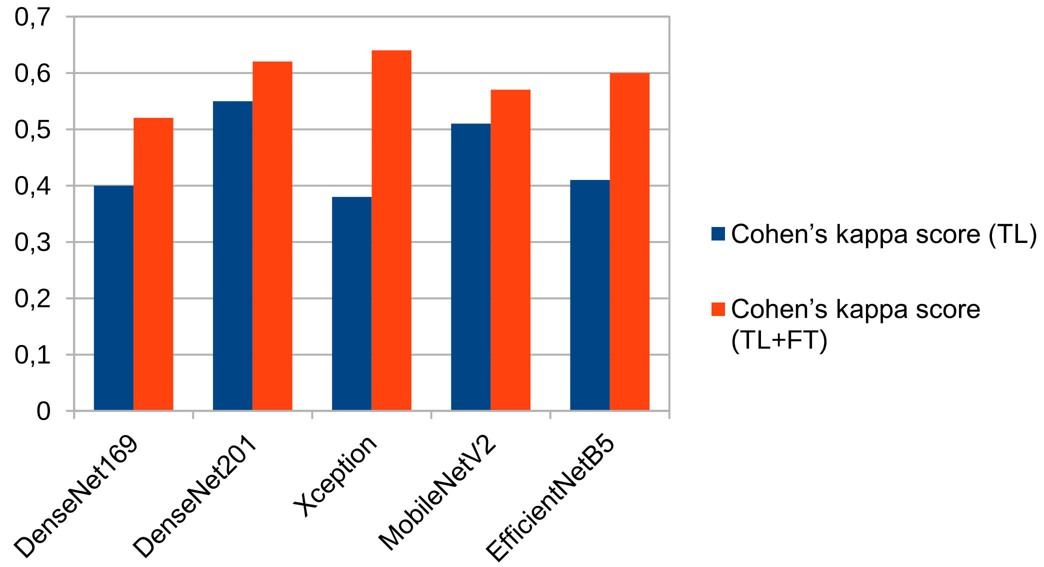


Figure 6.2. Comparison of the Cohen's kappa score obtained from Transfer learning and Fine-tuning method

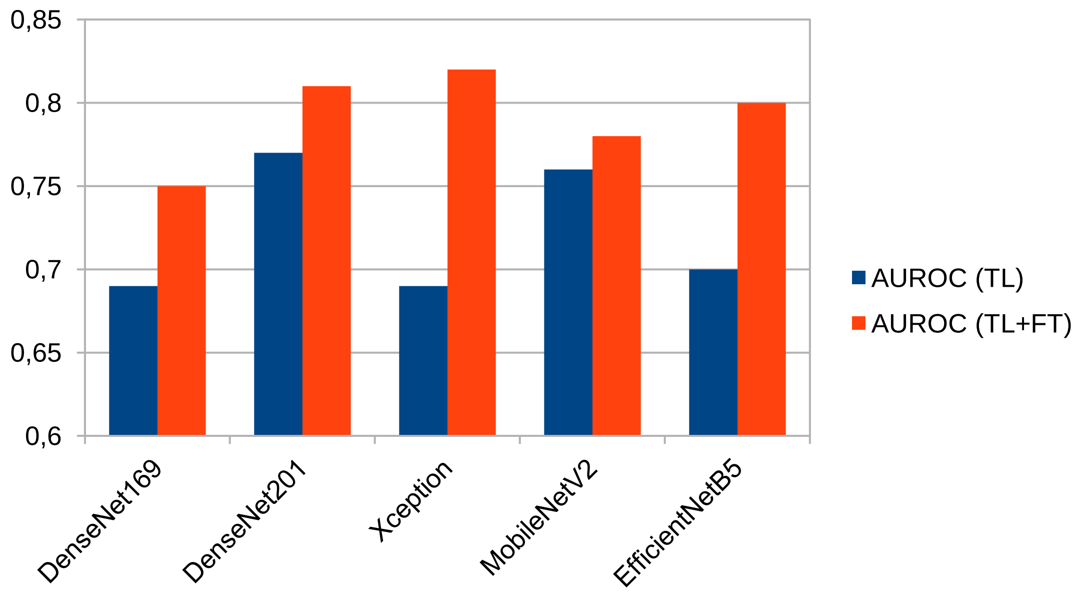


Figure 6.3 Comparison of the AUROC obtained from Transfer learning and Fine-tuning methods

## 6.2. Classification Results with Transfer learning and CLAHE

In this section, the classification results obtained after applying the CLAHE method to the X-ray images, are being discussed. The training was performed using dataset-1. The highest accuracy of 74%, the precision of normal class, recall of abnormal class, and F1-score of normal and abnormal classes were achieved using the EfficientNetB5 network. However, the results achieved by DenseNet-201 are comparatively good. It had the highest values for the precision of abnormal class, recall of normal class, and F1-score of normal class. All the results are shown in Table 6.7.

Table 6.7. Comparison of performance metrics of five models on test data after applying CLAHE (Dataset-1)

| **Architectures** | **Overall accuracy** | **Class** | **Precision** | **Recall** | **F1-score** |
|---|---|---|---|---|---|
| DenseNet169 | 0.67 | Normal | 0.64 | 0.86 | 0.73 |
| | | Abnormal | 0.74 | 0.45 | 0.56 |
| DenseNet201 | **0.70** | Normal | **0.66** | **0.91** | **0.76** |
| | | Abnormal | **0.81** | **0.46** | **0.59** |
| Xception | 0.61 | Normal | 0.61 | 0.74 | 0.67 |
| | | Abnormal | 0.61 | 0.46 | 0.52 |
| MobileNetV2 | 0.63 | Normal | 0.60 | 0.91 | 0.72 |
| | | Abnormal | 0.76 | 0.31 | 0.44 |
| EfficientNetB5 | **0.74** | Normal | **0.73** | **0.79** | **0.76** |
| | | Abnormal | **0.74** | **0.67** | **0.71** |

The performance of the models evaluated with Cohen’s kappa score and AUROC is best when the EfficientNetB5 network was trained as shown in Table 6.8 below.

Table 6.8 Cohen's kappa and AUROC results of five models after applying CLAHE (Dataset-1)

| **Architectures** | **Cohen's kappa score** | **AUROC** |
|---|---|---|
| DenseNet169 | 0.32 | 0.75 |
| DenseNet201 | 0.37 | 0.68 |
| Xception | 0.20 | 0.60 |
| MobileNetV2 | 0.23 | 0.61 |
| EfficientNetB5 | **0.47** | **0.73** |

In spite of that, the DenseNet-201 network achieved high accuracy when the CLAHE method was applied to the images. However, compared to the accuracies of Transfer learning a noticeable drop can be seen (Figure 6.4) except for the EfficientNetB5, which got better results.

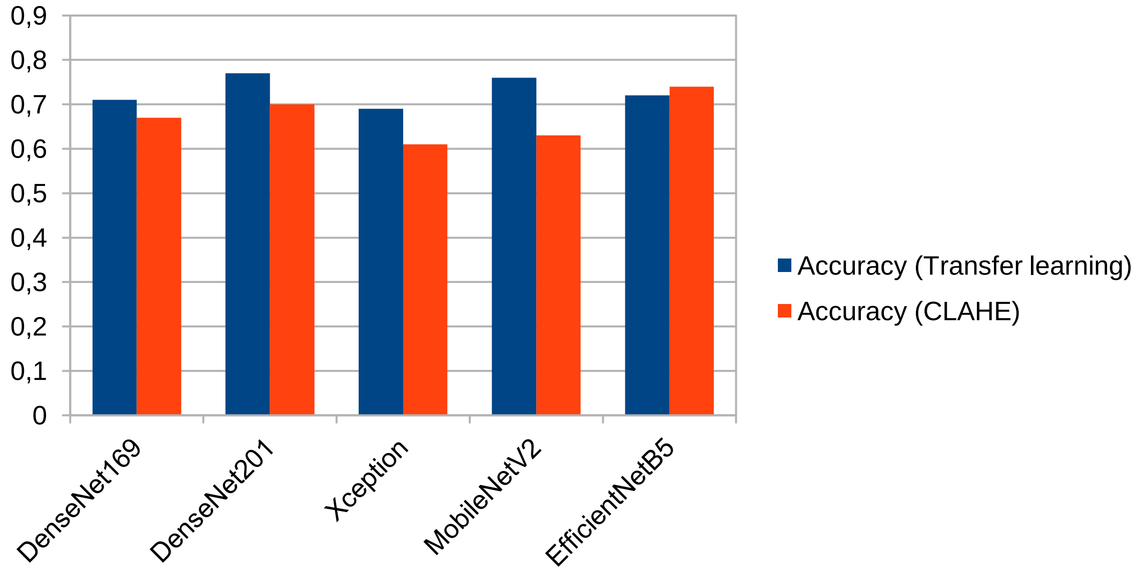


Figure 6.4 Comparison of the accuracies obtained from Transfer learning and CLAHE

The lower performance of the models can be seen from the values of the evaluation metrics Cohen's kappa score and AUROC as shown in Figure 6.5 and Figure 6.6

respectively. Considering the low results obtained, the CLAHE method was not used in the following experiments.

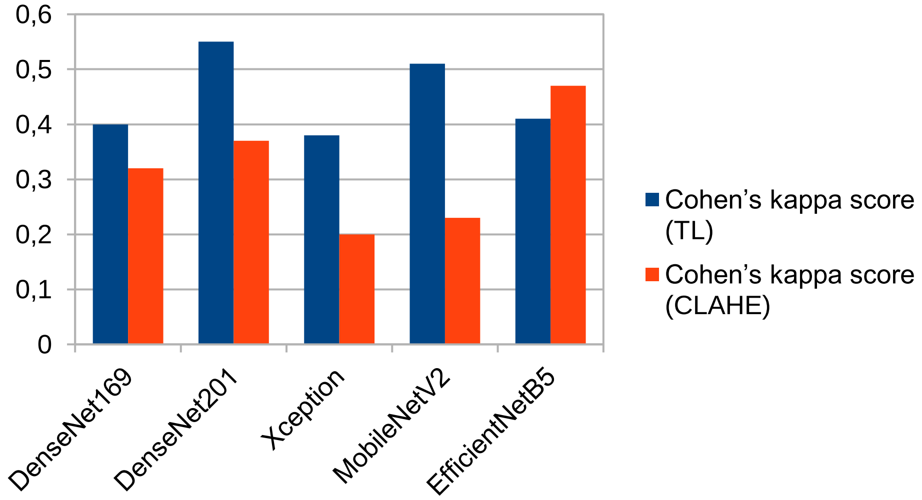


Figure 6.5 Comparison of the Cohen's kappa score obtained from Transfer learning and CLAHE method

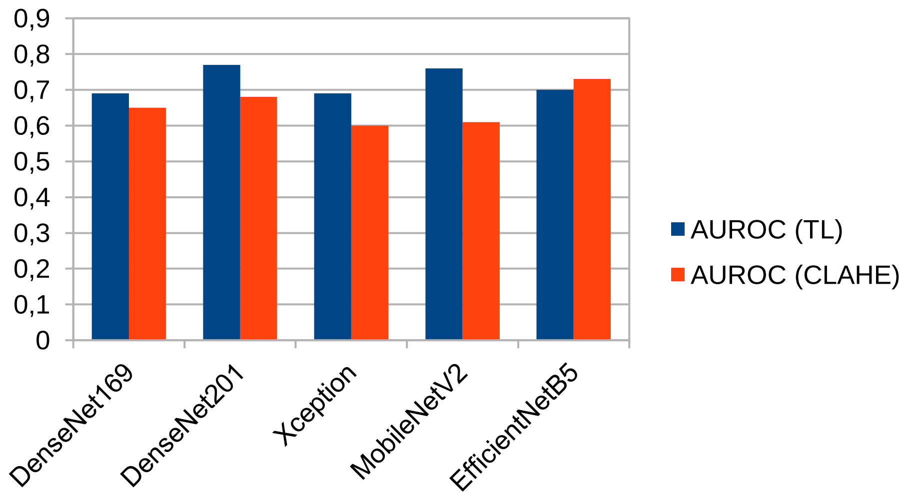


Figure 6.6 Comparison of the AUROC obtained from TL and CLAHE methods

## 6.3. Classification Results with Ensemble Learning

To evaluate the ensemble learner, performance analysis was made on the most up-to-date selected methods. Since there is an imbalanced problem in the dataset, it would not be correct to compare the achieved results based only on accuracy. Therefore, improvements on Cohen's kappa, AUROC, or F1 scores should be considered. The training was performed using dataset-2. The method of majority voting is used to gain the final predictions of the ensemble model. As it can be seen in Table 6.9, the overall accuracy has improved by 1% from the best model, though the other evaluation metrics show a better improvement in the ensemble learner.

Tablo 6.9 Comparison of the evaluation metrics for the Ensemble model (Dataset-2)

| **Architectures** | **Accuracy** | **Cohen's kappa score** | **AUROC** | **F1-score** |
|---|---|---|---|---|
| DenseNet201 | 0.82 | 0.62 | 0.80 | 0.76 |
| MobileNetV2 | 0.78 | 0.53 | 0.75 | 0.68 |
| ResNet50V2 | 0.81 | 0.59 | 0.78 | 0.74 |
| EfficientNetB4 | 0.81 | 0.60 | 0.78 | 0.74 |
| EfficientNetV2M | 0.80 | 0.58 | 0.78 | 0.74 |
| Ensemble (majority vote) | **0.83** | **0.63** | **0.80** | **0.76** |

## 6.4. Comparison with the Related Works

In this section, the results of different studies in the literature for the classification of the MURA images are mentioned and a comparison is made with the models proposed in this study on the accuracy metric. Table 6.10 shows the results of the models used to solve the binary classification (Normal/Abnormal) problem. The deep learning models, VGG16, DenseNet, and EfficientNet used in these studies are trained by transferring the weights previously obtained by the learning transfer method. The last layer of the aforementioned architectures was changed in accordance with the problem and used in the training. In these studies, dataset-1 was used, as it was given in the competition by Stanford ML Group for classifying musculoskeletal X-ray images. Considering the imbalance of the dataset, the results should not be compared only on accuracy. Hence, Cohen's kappa score is measured as it takes into account the imbalance in class distribution and removes the possibility of the classifier and a random guess agreeing. As it can be seen from the table, the proposed model using Transfer learning with Fine-tuning has achieved high accuracy of 86% as Solovoya's work, but higher Cohen's kappa score of 62% than the other related works.

Table 6.10 The accuracy rate results in different studies and the networks they used

| **Study** | **Network** | **Accuracy** | **Cohen's kappa score** |
|---|---|---|---|
| Rajpurkar et al. | DenseNet169 | x | 0.77 |
| Choudhary et al. | VGG16 | 0.76 | 0.53 |
| Solovoya et al. | DenseNet169 | 0.86 | x |
| Transfer learning (Dataset-1) | DenseNet201 | 0.77 | 0.55 |
| **Fine-tuning (Dataset-1)** | **DenseNet201** | **0.86** | **0.62** |
| TL+CLAHE (Dataset-1) | EfficientNetB5 | 0.70 | 0.47 |

As a second experiment, the dataset-2 was used with five networks and ensemble learning was applied. Even though the used dataset in He's work is different from dataset-2, a comparison can be made considering that ensemble learning was used, too. He et al. have used only two networks (DenseNet169, ResNet) with the ensemble learner, whereas in this work five networks were used (DenseNet201, ResNet50V2, MobileNetV2, EfficientNetB4, and EfficientNetV2M). In spite of that, according to Table 6.11, the results obtained in He's work are higher than the results obtained with our ensemble learner. An accuracy of 83% and a kappa score of 63% were achieved despite using the new dataset splitting which resulted in having 8001 test samples rather than 200 as used with He's ensemble learner. Considering the usage of the dataset-2, the proposed ensemble learner has achieved comparatively high accuracy and Cohen's kappa score.

Table 6.11 The accuracy rate results in different studies using Ensemble learning

| **Study** | **Network** | **Accuracy** | **Cohen's kappa score** |
|---|---|---|---|
| He et al. | Calibrated ensemble learner | 0.87 | 0.74 |
| Ensemble learner (Dataset-2) | DenseNet201, ResNet50V2, MobileNetV2, EfficientNetB4, EfficientNetV2M | 0.83 | 0.63 |

The achieved results in this work utilizing only one network for training are comparatively better than the results from the other related works. Although the accuracy from the proposed ensemble learner is lower than the accuracy achieved using one network, it can be said that it is a good result according to the usage of dataset-2.

# 7. CONCLUSION AND RECOMMENDATIONS

The main purpose of this study is the classification of musculoskeletal abnormalities by deep learning methods using musculoskeletal X-ray images. The problem is a binary classification of normal and abnormal X-ray images. Various Convolutional Neural Networks were pre-trained with different methods with the MURA dataset. To get the optimal parameters for each network a Hyperband optimization algorithm was used and according to the results, the networks were trained with the X-ray images. To achieve the aim of the study, two Transfer learning approaches were used. The first one was utilizing five different networks: DenseNet-169, DenseNet-201, Xception, MobileNet, and EfficientNet, to classify the images. The second approach was utilizing an ensemble model of five different networks: DenseNet-201, ResNet50V2, MobileNetV2, EfficientNetB4, and EfficientNetV2M. An additional experiment was used to see how it will reflect on the final results – the appliance of the CLAHE pre-processing method on the X-ray images. A set of various evaluation metrics were compared in this study to evaluate the performance of the models – accuracy, precision, recall, F1-score, Cohen’s kappa score, and AUROC. As a result of this work, the DenseNet-201 model reached the highest accuracy in the proposed methodology in the binary classification problem involving normal and abnormal classes and comparatively good results with the CLAHE method. The results are as follows: accuracy of 70% with the CLAHE method and accuracy of 86% with Fine-tuning method. Considering these results and comparing them to previous works, it can be seen that the performance of the model is higher than others. As a second experiment, the ensemble learner has also achieved very good results, having in mind the new dividing of the dataset. It managed to gain an overall accuracy of 83% from the combination of five models.

The aim of this study, which is focused on the classification of musculoskeletal X-ray images, is to help physicians diagnose abnormalities in the upper extremity and apply the required treatment. Following this study, a real-time mobile application that classifies X-ray images should be developed, specifically to help physicians performing emergency services. The present work can also be extended with the implementation of other advanced Deep Learning algorithms to achieve higher overall results.

## REFERENCES

Abumalloh, R., Nilashi, M., Ismail, V., Alhargan, A., Alghamdi, A., Alzahrani, A., Saraireh, L., Osman, R., Asadi, Sh., Medical image processing and COVID-19: A literature review and bibliometric analysis, 2021

Ajmera, P.; Kharat, A.; Botchu, R.; Gupta, H.; Kulkarni, V.; Real-world analysis of artificial intelligence in musculoskeletal trauma, 2021

Alzubaidi, L., Zhang, J., Humaidi, A., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M., Al-Amidie, M., Farhan, L., Review of deep learning: concepts, CNN architectures, challenges, applications, future directions,2021

Anu, T.C.; Mallikarjunaswamy, M.S.; Raman, R.; Detection of Bone Fracture using Image Processing Methods, 2015

Arias-Garzón, D.; COVID-19 detection in X-ray images using convolutional neural networks, 2021

Arslan,T.,(2015). X-Işınları Ve Kullanım Alanları,Erişim Tarihi : 10.05.2020
https://neu.edu.tr/wp-content/uploads/2015/11/BMT-211-ders-notu-X-Isinlari.pdf

Bhattacharya, S., Maddikunta, P., Pham, Q., Gadekallu, T., Krishnan, S., Chowdhary, C., Alazab, M., Piran, J., Deep learning and medical image processing for coronavirus (COVID-19) pandemic: A survey, 2020

Bhattacharya, S., Maddikunta, P., Pham, Q., Gadekallu, T., Krishnan, S., Chowdhary, Ch., Alazab, M., Piran, J., Deep learning and medical image processing for coronavirus (COVID-19) pandemic: A survey, 2020

Burden of Musculoskeletal Diseases in the United States, Third Edition, 2015

Chada, G.; Machine Learning Models for Abnormality Detection in Musculoskeletal Radiograph, 2019

Chauhan, T., Palivela, H., Tiwari, S., Optimization and fine-tuning of DenseNet model for classification of COVID-19 cases in medical imaging, 2021

Chollet, F., Xception: deep learning with depthwise separable convolutions, in: 2017 Proceedings of the IEEE conference on computer vision and pattern recognition, IEEE, 2017, pp. 1251 1258.

Choudhary, R.; Choudhary, S.; Meena, G., Abnormality Detection in Musculoskeletal Radiographs, 2020

Elton, R. C., X -Ray lasers, 1990, Boston, Academic Press

**REFERENCES (continued)**

Harel M, Mannor S. Learning from multiple outlooks. In: Proceedings of the 28th international conference on machine learning. 2011. p. 401–8.

Harini, N., Sriram, S., Sowmya, V., Soman, K.; Musculoskeletal radiographs classification using deep learning, 202

He, K., Zhang, X., Ren, Sh., Sun, S., Deep Residual Learning for Image Recognition, 2015

He, M., Wang, X., Zhao, Y.; A calibrated deep learning ensemble for abnormality detection in musculoskeletal radiographs, 2021

Huang, G., Liu, Z, Maaten, L., Densely Connected Convolutional Networks, 2018

Inik, Ö. & Ülker, E . (2017). Derin Öğrenme ve Görüntü Analizinde Kullanılan Derin Öğrenme Modelleri.

Lee, J., Jun, S., Cho, Y., Lee, H., Kim, G., Seo, J., Kim, N., Deep Learning in Medical Imaging: General Overview, 2017

Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., Talwalkar, A., Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization, 2018

Lo, W., Yang, X., Wang, Y., An Xception convolutional neural network for malware classification with transfer learning, 2019

Muniyappan, S., Allirani, A., A Novel Approach for Image Enhancement by Using Contrast Limited Adaptive Histogram Equalization Method, 2013

Naes, T., Kvaal., K., Artificial Neural Networks in Multivariate Calibration, 1993

Nam J, Kim S (2015) Heterogeneous defect prediction. In: Proceedings of the 2015 10th joint meeting on foundations of software engineering. 2015. p. 508–19.

Paranjape, B., Naik, A., Object Detection Using SSD MobileNet on Indian Dataset, 2022

Rajpurkar, P.; Irvin, J.; Bagul, A., MURA: Large Dataset for Abnormality Detection in Musculoskeletal Radiographs, 2018

Rawat, W., Wang, Z., Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review, 2017

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A., Fei-Fei, L., ImageNet Large Scale Visual Recognition Challenge, 2015

**REFERENCES (continued)**

Saif, A.; Shahnaz, C.; Zhu, W.; Ahmad, M.; Abnormality Detection in Musculoskeletal Radiographs Using Capsule Network, 2019

Sandler, M., Howard, A., Zhu, M., Zhu, A., Chen, L., MobileNetV2: Inverted Residuals and Linear Bottlenecks, 2019

Shorten, C., Khoshgoftaar, T., A survey on Image Data Augmentation for Deep Learning, 2019

Solovoya, A.; Solovyov, I., X-Ray bone abnormalities detection using MURA dataset, 2020

Sultan, H., Al-Atabany, W., Salem, N., Multi-Classification of Brain Tumor Images Using Deep Neural Network, 2019

Tan, M., Le, Q., EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,2020

Toğaçar, M., Ergen, B., Cömert, Z., Özyurt, F., A Deep Feature Learning Model for Pneumonia Detection Applying a Combination of mRMR Feature Selection and Machine Learning Models, 2019

Wang C, Mahadevan S. Heterogeneous domain adaptation using manifold alignment. In: Proceedings of the 22nd international joint conference on artificial intelligence, vol. 2. 2011. p. 541–46.

Weiss, K., Khoshgoftaar, T., Wang, D., A survey of transfer learning, 2016

Yoon, H., Jeong, Y., Kang, H., Jeong, J., Kang, D., Medical Image Analysis Using Artificial Intelligence, 2019

Zayegh, A., Albassam, N., Neural Network Principles and Applications, 2018

Zhou JT, Pan S, Tsang IW, Yan Y. Hybrid heterogeneous transfer learning through deep learning. In: Proceedings of the national conference on artificial intelligence, vol. 3. 2014. p. 2213–20.

Zhu Y, Chen Y, Lu Z, Pan S, Xue G, Yu Y, Yang Q. Heterogeneous transfer learning for image classification. In: Proceedings of the national conference on artificial intelligence, vol. 2. 2011. p. 1304–9.

Zou, K., O'Malley, A., Mauri, L., Receiver-Operating Characteristic Analysis for Evaluating Diagnostic Tests and Predictive Models, 2007