CHAPTER 1
INTRODUCTION

CHAPTER 1 INTRODUCTION

Currently image identification is the fastest developing technology for developers in the form of data. Let us take an example to better understand what image identification is. Today Google uses image captcha for validation of users. Nowadays in social media in which there are multiple images of users in the form tagged images or untagged images. So in social media this technology plays an vital role to identify users by their facts with 95% and above accuracy

Present computer vision is used in many fields like Vision biometrics, Object recognition, Social Media, Smart cars and many more. In many of these image recognition and classification plays a major role. In simpler terms, Image recognition is how well a computer can recognize objects, writings, actions, places. And image classification is classification of images based on the contextual information in image

Recently, image classification is growing and becoming a trend among technology developers especially with the growth of data in different parts of industry such as e-commerce, automotive, healthcare, and gaming. The most obvious example of this technology is applied to Facebook.

CBIR is a technique used for retrieval of images from a database for a specific user query. CBIR technique is used to fetch the images which the user actually wants to fetch based on different parameters.

About Image Classification Dataset

CIFAR-10 is a very popular computer vision dataset (Also CIFAR-100 is available for larger data) This dataset is well studied in many types of deep learning research for object recognition. This dataset consists of 60,000 images divided into 10 target classes, with each category containing 6000 images of shape 32*32. This dataset contains images of low resolution (32*32), which allows researchers to try new algorithms. The 10 different classes of this dataset are:

1. Airplane
2. Car

3. Bird
4. Cat
5. Deer
6. Dog
7. Frog
8. Horse
9. Ship
10. Truck

The CIFAR-10 dataset is already available in the dataset's module of Keras. We do not need to download it; we can directly import it from keras.datasets

1.1 SCOPE OF THE PROJECT

In the near future, the proposed model can also be developed to a neural network model that takes the values of different features as input for generating an accurate precise by considering the contextual meaning. To increase the accuracy of the precis, various features.

It has immense application in various domains :-

→ Biomedical and other healthcare applications

→ Disaster management

→ Automotive sector

→ Robotics

1.2. SOCIETAL / ENVIRONMENTAL IMPACT / NOVELTY OF IDEA

Identification of Images will be used to narrow the gap between computer vision and human vision so that machine can recognize the image in same way like human recognize a image

It will impact people's lives by reducing their time to get the details of Certain Objects & Images within no time. It will make human life easier.

GUI based demonstration of the algorithms and technique on Advanced CNN, ANN and Deep Learning used for Image Classification.

CHAPTER 2

PROBLEM DEFINITION

CHAPTER 2 PROBLEM DEFINITION

Identification of Images using CBIR with Keras is a problem of identifying and detecting real time images and Objects.

It is not easy to find and detect objects with high accuracy in various Domains(Automated Vehicles, medical imaging, object identification in satellite images, traffic control systems, brake light detection, machine vision, and more.)

Problem: “Detecting and Classifying Images & objects of real time data.”
Solution: CNN, Advanced CNN, CBIR

CHAPTER 3

LITERATURE REVIEW

CHAPTER 3 LITERATURE SURVEY

[1] Image Classification Using Deep Learning: -

The author briefly describes the process and the architecture that will be used to capture the image and later its features with the help of various algorithms used. Which goes like, it will first capture the image through a digital camera or else it will capture through Database. Every image will be normalized to predefined size for the further process. And further for Dimensionality reduction we use feature extraction methods like M-BTC(Block Transition Coding), Histogram Equalization,etc.

With the help of the image, feature vectors are created by extracting the feature by using different methods like MBTC(Block Transition Coding), Histogram Equalization,etc. This processed image will be given to the NN for the classification process.

Therefore with the help of tensorflow technique once we input the image, it traverses through n hidden layers, each responsible for performing specific operation, and finally produces the output at the output layer.
So, Currently this system focuses on creating only 4 classes namely (indoor, outdoor, cat, dog). The proposed system is developed using Python and Tensorflow framework for CPU based and using CUDA library for GPU based.

Further the author has described the future work, where several algorithms and different weight adjacent functions of deep learning will be considered in order to compare the performance enhancement with GPU Platform.

[2] Image classification using Deep learning: -

As of now, we are pretty much aware of the terms like image processing and neural networks. Moving forward, here the author has used an AlexNet architecture with convolutional neural networks for this purpose.

The working procedure described in this particular research paper is in such a way that four test images are selected from the ImageNet database for the classification purpose. Then the images are cropped having different portion areas and several experiments are conducted for the output purpose. Further the results show the effectiveness of deep learning based image classification using AlexNet.

[3] Research on image classification model based on deep convolutional neural network: -

As we know that image classification is one of the hot research directions in the computer vision field, and it is also the basic image classification system in other image application fields, which is usually divided into three important parts: image preprocessing, image feature extraction and classifier.

The researchers proposed an innovative training criterion of depth neural network for maximum interval minimum classification error. At the same time, the cross entropy and M3 CE are analyzed and combined to obtain better results. Finally, we tested our proposed M3 CE-CEc on two deep learning standard databases, MNIST and CIFAR-10. The experimental results show that M3 CE can enhance the cross-entropy, and it is an effective supplement to the cross-entropy criterion. M3 CE-CEc has obtained good results in both databases.

[4]Tensorflow Based Image Classification using Advanced Convolutional Neural Network : -

In this Research study image identifications will be done by the help of Advanced CNN (Convolutional Neural Networks with Tensorflow Framework. Here the author has used Python as a main programming language because Tensorflow is a python library. In this study input data mainly focuses on Plants categories by the help of leaves for identifications. Selecting CNN is the best approach for the training and testing data because it produces promising and continuously improving results on automated plant identifications. Here results are divided in terms of accuracy and time. Using advanced CNN results are above 95% while on others accuracy is below 90% and taking much more time than this.

In this whole framework the author described the process in such a way that there are 7 stages and each stage has their own discussion. All these stages depend on Tensor flow function which is an open source software and all the Tensorflow libraries are on python programming language and on importing Tensorflow each stage process will be done as per design.
The author concluded that, in this study they have performed classifications on leaves of plants by using CIFAR 10 dataset. As results we check the comparison between multiple models with specified dataset. All the results are achieved as per objectives by the Advanced CNN with accuracy of more than 95% while others are not capable of giving results as per objective. Advanced CNN is our main agenda for image classification because adding dense layers and increasing epochs gives desired results in a better way. Epochs are used to control the over fitting problems. Advanced CNN are very fast in comparison to others, it takes very less time for classification.

[5] Image Classification Using Machine Learning and Deep Learning Model : -

One of the major problems was that of image classification, which is defined as predicting the class of the image. Cat and Dog image classification is one such example of where the images of cat and dog are classified. This paper aims to incorporate state-of-art techniques for object detection with the goal of achieving high accuracy. A convolutional neural network has been built for the image classification task.

This study examined the efficacy of the pre-training Convolutional neural network with natural images of cat -dog dataset which has around 4000 training images for each of the categories. We achieved the best classification accuracy of 88.31 % with the optimal parameter settings of the proposed architecture.

Accuracy improvement can be done with the help of including more layers (streamlining the design) or gathering additional data and hyperparameters of the system.

[6] A study on Image Classification based on Deep Learning and Tensorflow

This research study about image classification by using the deep neural network (DNN) or also known as Deep Learning by using framework TensorFlow. The input data mainly focuses in flowers category which there are five (5) types of flowers that have been used in this paper.The roles of epochs in DNN was able to control accuracy and also prevent any problems such as overfitting.DNN has been choosing as the best option for the training process because it produced a high percentage of accuracy.

Implementation of deep learning by using framework TensorFlow also gave good results as it is able to simulate, train and classify with up to 90% percent of accuracy towards five (5) different types of flowers that have become a trained model. Lastly, Python has been used as the programming language throughout this research since it comes together with framework TensorFlow which leads to designing of the system involving Python from start until end.

[7] Image Classification Using CNN :-

This paper is published by **“Atul Sharma & Gurbakash Phonsa” in 2021.** In this paper, After using the CNN method, we were able to obtain 94 percent validation accuracy of our model. We trained and then tested the images and used the CNN method for the classification. This study focused on the CNN model and the accuracy of validation. After 20 epochs, the CNN model reached 90 percent accuracy of validation and then we loaded that model to label the images. We did the literature survey and found that different CNN techniques are being used to perform different operations and based on the computational power of any particular project or the complexity different methods of CNN can be compared.

[8] Image Classification using Convolutional Neural Networks :-

This paper is published by **“Muthukrishnan Ramprasath & M.Vijay Anand” in 2018.** In this paper, they have used Convolutional Neural Networks (CNN) for image classification using images from hand written MNIST data sets. These data sets were used both for training and testing purposes using CNN. It provides an accuracy rate of 98%. Images used in the training purpose are small and Grayscale images. The computational time for processing these images is very high as compared to other normal JPEG images. Stacking the model with more layers and training the network with more image data using clusters of GPUs will provide more accurate results of classification of images. The future enhancement will focus on classifying the colored images of large size and its very useful for image segmentation process.

CHAPTER 4

PROJECT DESCRIPTION

CHAPTER 4 PROJECT DESCRIPTION

We are going to train a Machine Learning model to learn differences between different categories of classes. The model will predict if a new unseen image is matched or any given image, basically CBIR technique.

The code architecture is robust and can be used to recognize any number of image categories, if provided with enough data. To learn about thousands of objects from millions of images, we need a model with a large learning capacity.

However, the immense complexity of the object recognition task means that this problem cannot be specified even by a dataset as large as ImageNet, so our model should also have lots of prior knowledge to compensate for all the data we do not have. (CIFAR-10 Dataset)

Thus, compared to standard feedforward neural networks with similarly sized layers, CNNs have much fewer connections and parameters and so they are easier to train, while their theoretically best performance is likely to be only slightly worse.

CHAPTER 5
REQUIREMENTS

CHAPTER 5 REQUIREMENTS

5.1. Functional Requirements

5.1.1. *User Interface*

5.1.1.1. The system should process the input given by the user only if it is an image file (JPG, PNG etc.)

5.1.1.2. System shall show the error message to the user when the input given is not in the required format.

5.1.1.3. System should identify the object present in the image.

5.1.1.4 System should be able to identify, match and produce the required image name, type or classification as per the database information.

5.1.2. *Training Data*

5.1.2.1. For training our system we will need a training data set.

5.1.3. *Testing Data*

5.1.3.1. After the model is trained, we will need a testing dataset for our model.

5.1.4. *Bandwidth*

5.1.4.1. The user should have a minimum of 2-5 Mbps speed to utilize the resource properly.

5.2. Non-Functional Requirements

5.2.1. *Reliability:*

5.2.1.1. This software will work reliably for low resolution images and not for graphical images.

5.2.1.2. Should be up for working as and when required.

5.2.2. *Usability:*

5.2.1.1. The application must be user friendly and easy to use.

5.2.1.2. The application must provide relevant information to its users.

5.2.3. *Availability:*

5.2.3.1. The application should be available whenever required.

5.2.3.2. This system will identify the object and match with the type or similar image which is present in the database.

5.3 Software/System Requirements

5.3.1. *Hardware Requirements*

5.3.1.1. Minimum 8 GB RAM

5.3.1.2. At Least 8 GB of available disk space.

5.3.2. *Software Requirements*

5.3.1.1. 64-bit Windows 8-11 / macOS 10.14 or higher / 64-bit Linux

5.3.1.2. Visual Studio Code

5.3.1.3. Python3.7 and above

5.3.3. *Bandwidth Requirements*

5.3.1.1. Bandwidth: 2-5 Mbps

CHAPTER 6

METHODOLOGY

CHAPTER 6 METHODOLOGY

With the rapid development of mobile Internet technology, more and more image information is stored on the Internet. Image has become another important network information carrier after text. Under this background, it is very important to make use of a computer to classify and recognize these images intelligently and make them serve human beings better.

What we see

0	0	0	0	0	0	0
0	1	0	0	0	1	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	1	0	0	0	1	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0

What a Computer see

Here CNN takes the most challenging task for identification of objects by using their complete picture or any parts of those objects while others tackle one by one process like firstly they take any specific organisms (color, shape, size etc.) then the whole picture of organisms. In CNN there are some limitations like it is not better with very large sets of images or lack of explanatory power. So Advanced CNN will replace CNN because Advanced CNN is small in size as compared to CNN for recognizing images. Here large models can be easily scaled up and these models are small enough to train fast, by this we will get out new ideas and have a good chance for experiment on other methods also. The architecture of Advanced CNN is multi-layer consisting of alternate use of Convolution layers and nonlinearities. All these layers are followed by fully connected layers leading into a softmax classifier. This model gives good accuracy results within a few times when we run on a GPU.

There are approximately between 10 epoch (each epoch 1563) operations with different modules independently in the whole training graph. There are generally three steps in training graphs:-

Model Inputs: Read operations and preprocess CIFAR images operations will be added for evaluation and training respectively.

Model Prediction: On supplied images classifications should be done by adding operations that perform inferences.

Model Training: Add operations that compute the loss, gradients, variable updates and visualization summaries.

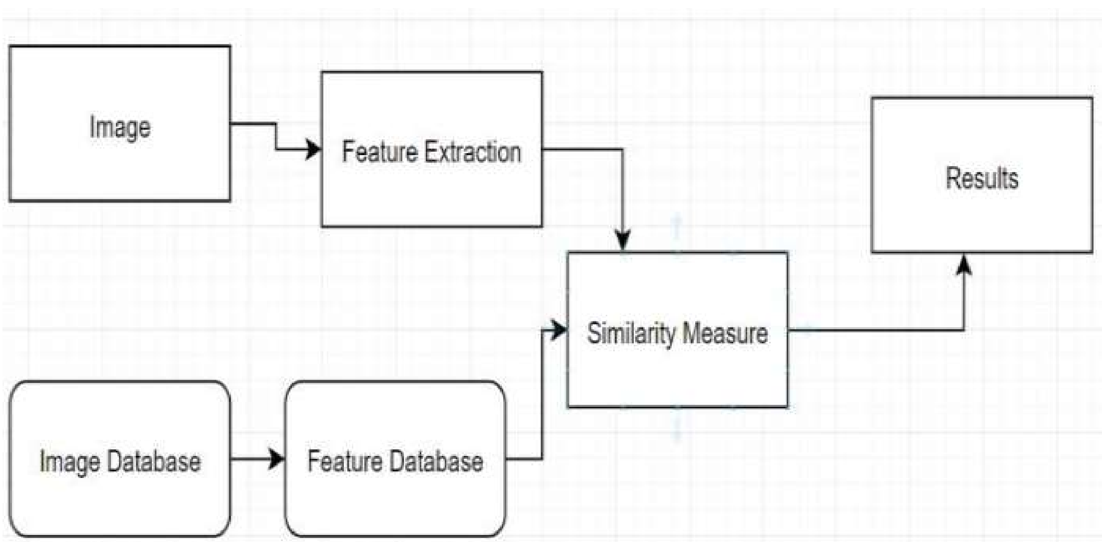


Figure 1. Image Extraction Process

CHAPTER 7

EXPERIMENTATION

CHAPTER 7 EXPERIMENTATION

1. Download and Prepare CIFAR-10 Dataset :-

In the CIFAR-10 dataset there are 60000 images which are in 10 classes and in each class there will be 6000 images. Here the dataset will be divided into two phases. First is training and second is testing phase.In training phase there are 50000 images while in testing phase there are 10000 images. Here all the classes are mutually exclusive and no overlapping between them.

2. Verification of Data :-

Verification of the dataset plays a vital role here. This verification will be done to check whether the given dataset is correct or not. For verification a few images will be used for plotting and these images are from the training phase and each image will be shown by their specific class name.

3. Create the Convolutional Base :-

Here Convolutional Base follows a common pattern in the form of Con2D and maxPooling2D layers in a stack. In this CNN takes input in the form of tensor shape (RGB) which means (image_height, image_width, colour_channels). As per above the output of Conv2D and maxPooling2D layers are in 3-D tensor shape. If you go deeper in the Network the height and width will be shrinked. The output of each Conv2D controlled by the arguments, if height and width is shrinked then we have to add more output channels in every Conv2D layers

4. Convert the pixel values of the dataset to float type and then normalize the dataset

```
train_x=train_X.astype('float32')
test_X=test_X.astype('float32')

train_X=train_X/255.0
test_X=test_X/255.0
```

5. Now perform the one-hot encoding for target classes

```
[5]
train_Y=np_utils.to_categorical(train_Y)
test_Y=np_utils.to_categorical(test_Y)

num_classes=test_Y.shape[1]
```

6. Create the sequential model and add the layers

```
model=Sequential()
model.add(Conv2D(32,(3,3),input_shape=(32,32,3),
    padding='same',activation='relu',
    kernel_constraint=maxnorm(3)))
model.add(Dropout(0.2))
model.add(Conv2D(32,(3,3),activation='relu',padding='same',kernel_constraint=maxnorm(3)))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
model.add(Dense(512,activation='relu',kernel_constraint=maxnorm(3)))
model.add(Dropout(0.5))
model.add(Dense(num_classes, activation='softmax'))
```

9. Train the model

```
model.fit(train_X,train_Y,
    validation_data=(test_X,test_Y),
    epochs=10,batch_size=32)
```

```
Epoch 1/10
1563/1563 [==============================] - 293s 186ms/step - loss: 1.7014 - accuracy: 0.3861 - val_loss: 1.4324 - val_accuracy: 0.4786
Epoch 2/10
1563/1563 [==============================] - 285s 182ms/step - loss: 1.3426 - accuracy: 0.5182 - val_loss: 1.1794 - val_accuracy: 0.5768
Epoch 3/10
1563/1563 [==============================] - 283s 181ms/step - loss: 1.1753 - accuracy: 0.5799 - val_loss: 1.1087 - val_accuracy: 0.6030
Epoch 4/10
1563/1563 [==============================] - 274s 175ms/step - loss: 1.0557 - accuracy: 0.6239 - val_loss: 1.0528 - val_accuracy: 0.6235
Epoch 5/10
1563/1563 [==============================] - 264s 169ms/step - loss: 0.9631 - accuracy: 0.6590 - val_loss: 1.0072 - val_accuracy: 0.6432
Epoch 6/10
1563/1563 [==============================] - 265s 170ms/step - loss: 0.8797 - accuracy: 0.6861 - val_loss: 0.9629 - val_accuracy: 0.6604
Epoch 7/10
1563/1563 [==============================] - 269s 172ms/step - loss: 0.8078 - accuracy: 0.7153 - val_loss: 0.9387 - val_accuracy: 0.6673
Epoch 8/10
1563/1563 [==============================] - 271s 173ms/step - loss: 0.7387 - accuracy: 0.7384 - val_loss: 0.9310 - val_accuracy: 0.6716
Epoch 9/10
1563/1563 [==============================] - 270s 173ms/step - loss: 0.6797 - accuracy: 0.7571 - val_loss: 0.9400 - val_accuracy: 0.6794
Epoch 10/10
1563/1563 [==============================] - 269s 172ms/step - loss: 0.6315 - accuracy: 0.7755 - val_loss: 0.9318 - val_accuracy: 0.6843
<keras.callbacks.History at 0x7f1f772769d0>
```

CHAPTER 8

TESTING AND RESULTS

CHAPTER 8 TESTING AND RESULTS

There are several tests conducted from this trained model. To prove trained model works, prediction done in a computer which is used in training using new test images Result model with lowest loss in validation is used (0.93) with accuracy 0.85 in validation images. There are several tests conducted from this trained model. To prove trained model works, prediction done in computer which used in training using new test images

```
1 model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 32, 32, 32)        896

 dropout (Dropout)           (None, 32, 32, 32)        0

 conv2d_1 (Conv2D)           (None, 32, 32, 32)        9248

 max_pooling2d (MaxPooling2D  (None, 16, 16, 32)       0
 )

 flatten (Flatten)           (None, 8192)              0

 dense (Dense)               (None, 512)               4194816

 dropout_1 (Dropout)         (None, 512)               0

 dense_1 (Dense)             (None, 10)                5130

=================================================================
Total params: 4,210,090
Trainable params: 4,210,090
Non-trainable params: 0
```

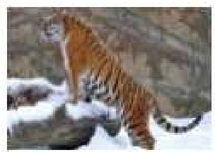
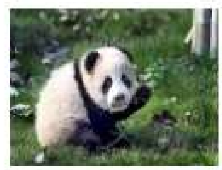
Sr. No	Actual Class	Image 1	Image 2	Image 3	Obtained Result Class
1	Zebra				Zebra
2	Tiger				Tiger
3	Panda				Panda
4	Horse				Horse

Figure 2. Obtained Result after classification

```
1 Image("ship.jpeg")
```

```
[44]  1 Image("horse.jpeg")
```

Figure 3. Some Sample Outcomes

CHAPTER 9

CONCLUSION AND FUTURE WORK

CHAPTER 9 CONCLUSION AND FUTURE WORK

9.1. Conclusion

In conclusion, in this paper, we proposed a CBIR system that uses CNNs to extract features from images and Machine Learning Techniques to retrieve similar images. Our experiments showed that the proposed CBIR system performs well on a large dataset of images and outperforms existing CBIR systems in terms of accuracy and efficiency. The proposed system can be extended to other datasets and can be used in various applications such as image retrieval, image search engines, and multimedia databases.

9.2. Scope For Future Work

As a future work, we will consider the following steps :

Integration with Cloud-based Services: Integration with cloud-based services such as AWS, Azure, or Google Cloud can be done to improve the performance and scalability of the system. These services provide a scalable and reliable infrastructure for hosting the application and processing large amounts of data.

Mobile application: Developing a mobile application for image identification can be a potential direction to take this project. This app can be used by users to identify images from their phones or tablets, and can be integrated with the backend server to provide quick and accurate results.

Integration with Social Media Platforms: The project can be extended to integrate with social media platforms like Instagram, Facebook, and Twitter. This can be achieved by developing an API that can fetch images from these platforms, and using the CBIR system to identify them.

Real-time Image Identification: Developing a real-time image identification system can be a challenging but exciting task. This system can be used in various applications like security, surveillance, and healthcare. The system can use advanced computer vision techniques to identify images in real-time and alert the relevant authorities if necessary.

CHAPTER 10 REFERENCES

[1] Dr. Vinayak Bharadi, Arusa Irfan Mukadam, Misbah Naimuddin Panchbhai, Nikita Narayan Rode (2017) “Image Classification Using Deep Learning” *International Journal of Engineering Research & Technology (IJERT)* Vol. 6 Issue 11, November - 2017

[2] Venu Gopala Rao Matcha, Harshali Mane (2018) “Image classification using Deep learning” *International Journal of Engineering & Technology · March 2018.*

[3] Mingyuan Xin and Yong Wang ``Research on image classification model based on deep convolutional neural network” *EURASIP Journal on Image and Video Processing (2019).*

[4] Pradumn Kumar, Upasana Dugal “Tensorflow Based Image Classification using Advanced Convolutional Neural Network” *International Journal of Recent Technology and Engineering (IJRTE) ,Volume-8 Issue-6, March 2020*

[5] K R Sinchana, Suman R Kunte “Image Classification Using Machine Learning and Deep Learning Model”

[6] Abu, M. A., Indra, N. H., Rahman, A. H. A., Sapiee, N. A., & Ahmad, I. (2019). A study on Image Classification based on Deep Learning and Tensorflow. *International Journal of Engineering Research and Technology*, *12*(4), 563-569.

[7] Sharma, A., & Phonsa, G. (2021, April). Image Classification Using CNN. In *Proceedings of the International Conference on Innovative Computing & Communication (ICICC)*.

[8] Ramprasath, M., Anand, M. V., & Hariharan, S. (2018). Image classification using convolutional neural networks. *International Journal of Pure and Applied Mathematics*, *119*(17), 1307-1319.

[9] Gross, J., Breitenbach, J., Baumgartl, H., & Buettner, R. (2021). High-performance detection of corneal ulceration using image classification with convolutional neural networks.

[10] Juneja, S., Juneja, A., Dhiman, G., Behl, S., & Kautish, S. (2021). An approach for thoracic syndrome classification with convolutional neural networks. *Computational and Mathematical Methods in Medicine*, *2021*.

[11] Algan, G., & Ulusoy, I. (2021). Image classification with deep learning in the presence of noisy labels: A survey. *Knowledge-Based Systems*, *215*, 106771.

[12] Yim, J., Ju, J., Jung, H., & Kim, J. (2015). Image classification using convolutional neural networks with multi-stage features. In *Robot Intelligence Technology and Applications 3* (pp. 587-594). Springer, Cham.

[13] Dubey, S. R. (2021). A decade-long survey of content-based image retrieval using deep learning. IEEE Transactions on Circuits and Systems for Video Technology, 32(5), 2687-2704.

[14] Shanmugamani, R. (2018). Deep Learning for Computer Vision: Expert techniques to train advanced neural networks using TensorFlow and Keras. Packt Publishing Ltd.

[15] Kumar, A., Singh, K. P., Kumar, S., & Vetrivendan, L. (2022). Image Classification in Python Using Keras. In Proceedings of Data Analytics and Management: ICDAM 2021, Volume 1 (pp. 541-556). Springer Singapore.

[16] Ali, M. (2023). Content Based Medical Image Retrieval: A Deep Learning Approach (Doctoral dissertation, St. Mary's University).

[17] Choe, J., Hwang, H. J., Seo, J. B., Lee, S. M., Yun, J., Kim, M. J., ... & Kim, B. (2022). Content-based image retrieval by using deep learning for interstitial lung disease diagnosis with chest CT. Radiology, 302(1), 187-197.