Explainable AI for Medical Imaging: A Deep Learning Approach towards Classification of Cervical Cancer

A thesis

Submitted in partial fulfillment of the requirements for the Degree of

Bachelor of Science in Computer Science and Engineering

Submitted by

Afsara Tasnim	**180104141**
Jakia Sultana Juthi	**180204030**
Ashfiqun Mustari	**180204067**
Rushmia Ahmed	**180204080**

Supervised by

Mr. G. M. Shahariar

Department of Computer Science and Engineering

Ahsanullah University of Science and Technology

Dhaka, Bangladesh

May 2023

CANDIDATES' DECLARATION

We, hereby, declare that the thesis presented in this report is the outcome of the investigation performed by us under the supervision of Mr. G. M. Shahariar, Department of Computer Science and Engineering, Ahsanullah University of Science and Technology, Dhaka, Bangladesh. The work was spread over two final year courses, CSE4100: Project and Thesis I and CSE4250: Project and Thesis II, in accordance with the course curriculum of the Department for the Bachelor of Science in Computer Science and Engineering program.

It is also declared that neither this thesis nor any part thereof has been submitted anywhere else for the award of any degree, diploma or other qualifications.

Afsara Tasnim
180104141

Jakia Sultana Juthi
180204030

Ashfiqun Mustari
180204067

Rushmia Ahmed
180204080

CERTIFICATION

This thesis titled, **"Explainable AI for Medical Imaging: A Deep Learning Approach towards Classification of Cervical Cancer"**, submitted by the group as mentioned below has been accepted as satisfactory in partial fulfillment of the requirements for the degree B.Sc. in Computer Science and Engineering in May 2023.

Group Members:

Afsara Tasnim	**180104141**
Jakia Sultana Juthi	**180204030**
Ashfiqun Mustari	**180204067**
Rushmia Ahmed	**180204080**

Mr. G. M. Shahariar
Lecturer & Supervisor
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

Dr. Md. Shahriar Mahbub
Professor & Head
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

ACKNOWLEDGEMENT

We would like to express our heartfelt gratitude and appreciation to Mr. G. M. Shahariar, our dedicated supervisor, for his invaluable guidance, support, and encouragement throughout the completion of our thesis on Explainable AI for Medical Imaging: A Deep Learning Approach towards Classification of Cervical Cancer. His expertise, insightful feedback, and continuous guidance have been instrumental in shaping this research work. We are also grateful to our exceptional teammates who have been a constant source of inspiration and collaboration. Their contributions, insightful discussions, and collective efforts have significantly enhanced the quality of this project. Their commitment and dedication to our common goal have made this journey a memorable one. Last but not least, We are grateful to Almighty Allah for bestowing us with the strength, wisdom, and determination to undertake this research and our family and friends for their understanding, and encouragement throughout this journey. Their constant support and belief in our abilities have been our pillars of strength. Once again, we express our deepest appreciation to our supervisor, and teammates for their invaluable contributions to this thesis. Their support and guidance have been integral to the successful completion of this research work.

Dhaka
May 2023

Afsara Tasnim

Jakia Sultana Juthi

Ashfiqun Mustari

Rushmia Ahmed

ABSTRACT

Cervical cancer is a significant public health concern affecting women of reproductive age worldwide. Although cytology-based screening programs exist to detect cervical cancer cells, the analysis of these cells by expert pathologists can be prone to errors. To address this issue, this study proposes an efficient system for classifying cervical cancer cells using deep learning models. The research begins by training five different deep learning models using a diverse dataset. In order to account for the varying costs associated with misclassifying different types of cervical cancer cells, the models are made cost-sensitive. This means that the models are trained to prioritize accurate classification of cells that are more critical to detect. To further enhance the performance of the models, supervised contrastive learning is included. This technique aims to improve the classification of cervical cancer cell images by learning to distinguish between positive and negative instances. By incorporating supervised contrastive learning, the models become more adept at capturing important features and patterns that classifies cervical cancer cells. Extensive experiments are conducted to evaluate the proposed methods using the SIPaKMeD dataset, which is a well-established benchmark in cervical cancer research. The results of the experiments demonstrate the effectiveness of the developed system, achieving an accuracy of 97.29%. Moreover, the research team recognizes the importance of building trust in the automated medical image classification system. To address this, 2 types of explainable artificial intelligence techniques are employed which are Gradient-Based and Perturbation-Based methods. These techniques enable the interpretation of how models come to their decisions, allowing medical professionals to understand and validate the classification results. By providing transparency and interpretability, the system aims to foster greater trust paving the way for improved cervical cancer detection and diagnosis. Our dataset and the implementation of all experiments can be found at https://github.com/isha-67/CervicalCancerStudy

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Cervical cancer, the world's third-most common type of cancer, is the leading cause of cancer-related deaths in women [1]. It is one of the most frequently occurring cancers in women and all women who are of reproductive age can be at risk of cervical cancer. Cervical cancer has grown to be such a major public health concern since it claims the lives of numerous women each year.

However, unlike other cancers, cervical cancer can be prevented. Many cytology-based screening programs can detect any form of abnormality in the cervical cells before it transforms into cancer. But screening programs that are based on cytology are difficult to implement, resulting in a significant number of deaths due to cervical cancer every year. One of the most popular screening tests for cervical cancer is the Papanicolaou test [2], which is also known as a Pap test or Pep smear. It is a test that can show anomalies in the cervical cells that can later develop into cancer. But one of the challenges of detecting cancerous cells from pap smear images is that it requires highly qualified pathologists to analyze the results, which might be hard to find, especially in developing countries. This is where computer-supported tools, such as deep learning, can be used to identify patterns relevant to medical diagnosis to make up for these limitations. Deep learning can be an important tool to recognizing patterns of malignancy, detecting any abnormality in the cervical cells, and perform image classification.

Image classification is generally a process of assigning a pre-defined label or a class to an image based on some specific features. Image classification takes an image as input, and then returns a class as an output. Based on the number of classes, image classification can be categorized as (i) Binary Classification and (ii) Multiclass classification. In Binary classifications, there are only two classes. Binary classification in image classification is a process where the input images can be classified in either of the two classes, for example, classifying images of humans as male or female.

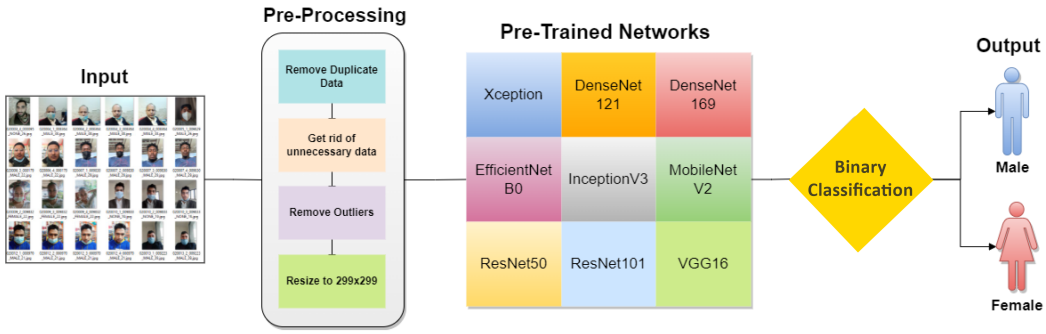


Figure 1.1: Binary Classification

On the other hand, in multiclass classification, there are usually a minimum of more than two classes to a maximum n number of classes. In a multiclass classification of images, the input images can be classified in any of the n numbers of classes- for example, recognizing a digit from images of handwriting.

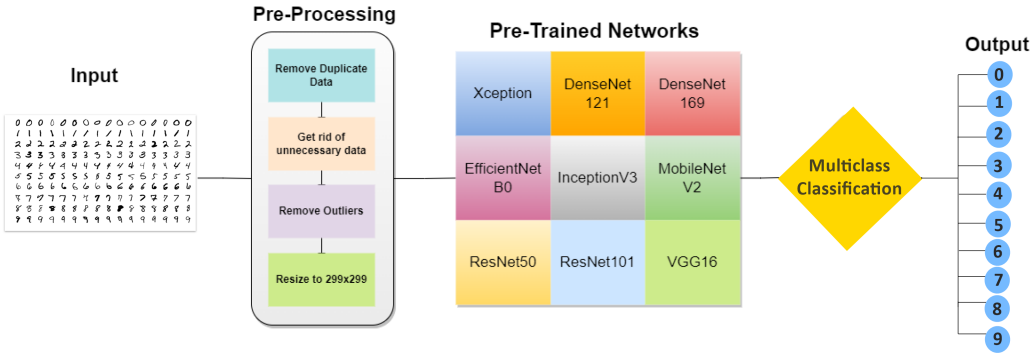


Figure 1.2: Multiclass Classification

Hence, medical image classification takes images from medical tests and then classifies the images into a category or class from a variety of classes to help doctors diagnose diseases. Image classification generally consists of two steps. The first step is known as feature extraction, which involves extracting certain features that will effectively help in classifying the images. The next step involves building models using the selected features and classifying the images into different classes. An image can be categorized into potentially n different classes. Usually, professional doctors perform feature extraction and image classification on medical images manually [3], which is very time-consuming and difficult, especially when there are a large number of images. Therefore, automating the entire process can be very helpful. This is where deep learning comes in.

Deep learning, also known as deep structured learning is a very efficient tool for large data analysis. It trains computers and machines to learn from experience, recognize patterns, and classify images similar to how the human brain does by using algorithms and artificial neural networks. But as opposed to traditional machine learning techniques, deep learning does

not require manual feature extraction, which, as a result, provides higher efficiency. Deep learning is carried out by neural networks, which consist of a large number of neurons as the human brain does. The input of the following layer is regarded as the output of the upper layer in neural networks since each layer contains a large number of neurons. Through links between layers, the neural network can change the initial input into the desired result. Deep learning overcomes the problem that machine learning necessitates manually extracting feature data by automatically acquiring more general and abstract features from the input. In deep learning, a very popular type of artificial neural network that can classify images more accurately is CNN, also known as a convolutional neural network.

CNNs or convolutional neural networks mimic how the human brain functions. Similar to how a neuron in the brain does when distributing information throughout the body, artificial neurons, called nodes in CNNs, receive inputs, analyze them, and send the outcome as output. The input of the convolutional neural network is the image. CNNs may have a number of hidden layers that employ mathematics to extract information from the image. Examples of this include convolutions, pooling, and fully connected layers. Convolution is the initial layer to extract necessary features from an input image. Going through the fully connected layers of the CNN model, the output layer recognizes the image and assigns a class to it. As convolutional neural networks are capable of classifying images without the need of extracting features and provide more accurate results than traditional neural networks, they have been used greatly for medical image classification.

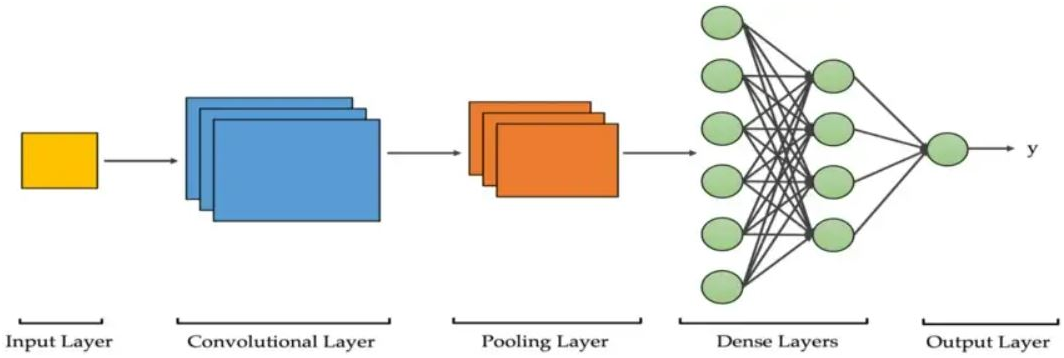


Figure 1.3: Diagram of CNN

There are a significant number of studies that have used deep learning, and convolutional neural networks (CNN) to classify and detect cervical cancer. Some of these studies experimented with different datasets, and some experimented with different kinds of convolutional neural networks. In a study conducted in 2021 [4], Tripathi et al. classified cervical cancer using deep learning algorithms. In another study, Pramanik et al. [5] classified cervical cells using the ensemble method. But a great concern associated with these studies is that how their models make such decisions to classify the images cannot be explained. Studies conducted by Panwar et al [6], and Esmaeili et al [7], show that they have performed

image classification using convolutional neural networks, and they have also interpreted how their models have come to such a conclusion with Explainable Artificial Intelligence, also known as XAI.

Explainable AI is a form of artificial intelligence that can be understood by humans. It provides a visual interpretation and explanation of how a model makes a decision about classifying an image based on its features. As deep learning is playing a significant role in medical image classification, explainable AI can provide an explanation for a patient's diagnosis. It can assist doctors in explaining to patients their diagnosis and how a treatment plan would benefit them. As a result, explainable artificial intelligence can be very helpful in the field of medicine when combined with medical imaging data to detect cancer.

In our study, we have taken the SIPaKMeD Dataset which consists of images from the Pap Smear test that have been manually defined by experts [8]. We develop a deep learning-based detection system that can determine the type of cancer cells. We fine-tune 5 different classifiers and make them cost-sensitive. We also introduce supervised contrastive learning [9] to experiment with different loss functions. We further interpret the classifier models with the help of explainable artificial intelligence to help people understand the underlying reasons behind our models to come to such a decision about classifying an image and build greater trust with automated medical image classification systems.

The study follows a structured format consisting of eight chapters. Chapter 1 serves as the introduction, providing an overview of the research topic and its significance. Chapter 2 focuses on related works, reviewing and discussing previous studies in the field. Chapter 3 provides a background study, including the theoretical foundations and concepts relevant to the research. Chapter 4 is dedicated to the dataset. Chapter 5 outlines the methodology employed in the research. Chapter 6 covers the evaluation, including evaluation metrics, hyper-parameter settings, experimental results, and interpretation using XAI. Chapter 7 compares the current research with previous works and discusses limitations and future directions. Finally, chapter 8 concludes the study, summarizing the findings and presenting the overall conclusions. The thesis book concludes with a references section listing the sources cited throughout the research.

Chapter 2

Related Works

Cervical cell classification can aid in the early detection of malignant subjects, which is a crucial step in the fight against cervical cancer. That's why a significant number of studies have classified cervical cancer in recent years. Some of these studies experimented with different datasets, and some experimented with different algorithms of machine learning and deep learning. To classify with better accuracy and improve models' generalization ability, different types of approaches have been adopted in the studies that are (1) traditional or deep learning, (2) fine-tuning on pre-trained models, (3) ensemble learning, and (4) feature fusion technique. An overview of the utilization of pre-trained convolutional neural network(CNN) architectures to classify cervical cells is compiled in this section.

Ghoneim et al. [10] presented a cervical cancer cell detection and classification system based on convolutional neural networks (CNN). The cell images were fed into a CNN model to extract deep learned features. The input images were then classified by an extreme learning machine-based classifier. The CNN model was subjected to transfer learning and fine-tuning techniques. Alternatives to the ELM, MLP, and autoencoder-based classifiers were also investigated. This experiment made use of the Herlev database. They investigated two deep CNN models in the proposed system: the VGG-16 Net and the CaffeNet. In the 2-class problem, the proposed system with the ELM-based classifier achieved 99.7% accuracy, and in the 7-class problem, it achieved 97.2% accuracy.

Rahaman et al. [11] introduced a hybrid deep feature fusion (HDFF) technique for enhancing the classification performance of cervical cell images. This method integrates feature vectors extracted from four different deep learning models, namely VGG-16, VGG-19, ResNet-50, and XceptionNet, to capture more significant information. The late fusion (LF) approach and basic DL models were compared with the author's suggested method. Late fusion (LF), uses the most classifier decisions possible and then weights each one to enhance classification performance. The publicly available SIPaKMeD dataset is used to test the suggested approach. The SIPaKMeD dataset includes 4049 pictures of cervical cells with

annotations. Each class uses 60% of the dataset for training, 20% for validation, and 20% for testing. In the preprocessing step, for all four CNN networks, they rescaled the object size to (224 x224) pixels and two steps of data augmentation tasks were conducted. After preprocessing, These pre-trained models are fine-tuned on the SIPAKMED dataset. They classified the dataset into five classes, three classes, and two classes. For binary-class, three-class, and five-class classification tasks, the HDFF method achieves maximum classification accuracies of 99.85%, 99.38%, and 99.14% among the basic DL model, LF, and HDFF approaches. Additionally, the authors also used the Herlev dataset containing 917 single-cell images to evaluate the performance of the proposed HDFF approach. The HDFF method achieved the highest classification accuracies of 98.91% and 90.32% on binary class and 7-class classification tasks, respectively.

Pramanik et al. [5] introduced an innovative ensemble approach focused on minimizing the error value between observations and ground truth. The fuzzy ensemble method's primary objective is to measure the significance of the classification models' confidence scores in various distance spaces. Evaluation of the proposed model was performed using the publicly available SIPAKMED dataset. This dataset consists of 4,049 images. The preprocessing stage involves resizing the training samples and applying online augmentation methods to enhance the images. These enhanced images are passed to three pre-trained ImageNet CNN models with extra layers to finetune the models. An ensemble approach is used to aggregate the confidence scores that are extracted from trained CNN models. They compared their method with inception V3, Inception ResNet V2, and MobileNet V2 models to demonstrate the superior achievement of the proposed method. They used 5-fold cross-validation to evaluate techniques on the SiPakMed dataset. With 96.96% accuracy, the proposed technique outperformed each of the three CNN models separately.

A classification model based on ensemble methods was introduced by Manna et al. [12] and used three different CNN architectures: Inception v3, DenseNet-169, and Xception. The proposed ensemble method increases classification accuracy by combining multiple CNN models that have been trained on cervical cytology images. The goal of this study was to combine multiple CNN models to enhance the classification accuracy of cervical cytology images. The proposed ensemble method made use of the collective decision-making abilities of various models rather than solely relying on a single CNN model. To capture various representations of the cervical cytology data, a collection of CNN models is trained using various architectures, hyperparameters, or initializations. The proposed model has been evaluated using a 5-fold cross-validation strategy on two benchmark datasets: the SIPaKMeD and the Mendeley LBC dataset. The SIPaKMeD dataset, which includes a collection of 4049 images, was the dataset used in this study. Five different classes contain an uneven distribution of these images. The Mendeley LBC dataset, which consists of 963 images that are similarly unevenly distributed among four different classes is also included. On the SiPaKMeD

dataset, they reported an accuracy of 98.55% for a 2-class classifier and 95.43% for a 5-class classifier. On the Mendeley LBC dataset, the model also achieved a remarkable accuracy of 99.23%.

Tripathi et al. [4] presented deep learning techniques to classify cells based on different stages of cancer cell growth. For the classification job, they employed ResNet-50, ResNet-152, VGG-16, and VGG-19 pre-trained models. The SIPaKMeD dataset, which includes a collection of 4049 images, was the dataset used in this study.The dataset was divided into training, validation, and testing sets in a ratio of 60,20, and 20, respectively. The authors fine-tuned the weights of the pre-trained models and conducted a comparative analysis of all the models. Upon observing the performance, ResNet-152 was elected superior with 94.89% accuracy.

Hsieh et al. [13] proposed detecting bone metastases on bone scans using image classification and contrastive learning. The dataset includes 37,427 sets of images from 19,041 patients. In order to identify bone metastases on whole-body bone scans, this study used a CNN-based architectures, DenseNet121 and ResNet50, to compare the contrastive learning approach . The outcome of the study show that contrastive learning may be used to medical functional pictures and it is beneficial in enhancing the accuracy of deep learning models. In this study, the accuracy of the ResNet50 model had an 94.30%, while the DenseNet121 model had an accuracy of 93.39%.

Ravi et al. [14] developed an attention-cost-sensitive deep learning-based feature fusion ensemble meta-classifier technique for skin cancer classification using the HAM10000 dataset . It is one of the widely used datasets that is used in many studies to compare the effectiveness of deep learning models for the identification and categorization of skin diseases. This study employs CNN-based pretrained models such as VGG16, MobileNet, InceptionV3, ResNet50,EfficientNetV2 model for skin cancer classification. To deal with the data imbalance during training, cost weights are included in the deep learning models. Furthermore, attention is also incorporated in order to extract the optimal features for accurately detecting skin diseases. The result of this experiment reveals that the, EfficientNetV2 model is performing better than the other model with an accuracy of 96%.

Panwar et al. [6] proposed a deep transfer learning algorithm that uses X-ray and CT-scan images of the chest to detect COVID19 cases more quickly.Because the available dataset of radiology imaging related to X-rays of COVID-19 patients is limited, they used CNN and VGG-19 as one transfer learning technique. The proposed model was 95.61% accurate. In all of the experiments, they used the GradCAM-based color visualization approach to clearly interpret radiology image detection. GradCAM is applied to any of the convolutional layers after the predicted label for the entire model has been calculated. In general, the final convolutional layer is considered the layer to be used for Grad-CAM.

Chapter 3

Background Study

In this chapter, a detailed background study is presented, focusing on the theoretical foundations and relevant concepts crucial to the research. The chapter covers several key topics, including the exploration of five pre-trained Convolutional Neural Networks (CNNs), namely VGG16, VGG19, DenseNet169, MobileNetV2, and ResNet50. These CNN architectures are widely used in computer vision tasks and provide a strong basis for image recognition and analysis. Additionally, the chapter discusses the concept of Cost-Sensitive Learning, which involves assigning different costs or weights to different classes. Furthermore, the study discusses Supervised contrastive Learning, which is a technique used to learn meaningful representations by contrasting positive and negative pairs of samples. Lastly, the chapter explores Explainable Artificial Intelligence (XAI), which aims to develop models and techniques that can provide human-interpretable explanations for the decisions and predictions made by classifiers.

3.1 Pre-trained Convolutional Neural Network

Convolutional neural networks (CNNs) that have already been trained on large- scale dataset using millions of images are deep learning models. These models can extract complex information from images, such as forms, and textures, which can be applied to tasks of computer vision, such as object recognition, classification, and image segmentation. Pre-trained CNNs can attain state-of-the-art performance with less training data and quicker training timeframes because they are often fine-tuned on a smaller dataset with a specified task.

3.1.1 ResNet50

The ResNet-50 (residual neural network) architecture was proposed by He et al. [15] as a variation of the ResNet architecture, which is a popular neural network that forms the basis for numerous computer vision applications. It is based on the nearly 1.2 million images in the ImageNet database, whose features and weights are carried over to the following task using the pretrained network. The main issues with deep networks are training difficulty and vanishing gradient problem, which makes learning insignificant at the initial layers in the backpropagation step. To prevent information loss during deep network training, the ResNet architecture employs the skip connections technique. Due to the existence of skip-connections, any architectural layer that negatively affects the performance of the model is skipped. Very deep networks can be trained using the skip connection technique, which can improve the model's performance. Figure 3.1 below depicts the skip connection:

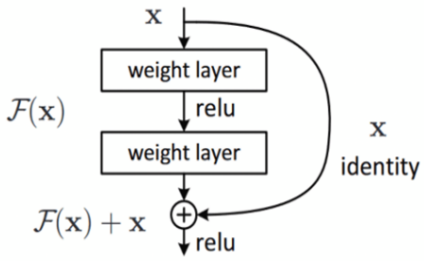


Figure 3.1: Skip Diagram of ResNet50

Here, F is the activation function. The result is F(w*x + b) = F(X). However, the output is F(X)+x with skip connection. The Resnet-50 architecture contains the following element: There are five convolutional layers in the ResNet-50 architecture. After loading the input image, then it is passed through a convolutional layer with a 64 -filter and a 7x7 kernel before being passed through a max pooling layer with a 2-stride length. After that, we have a 1x1 convolutional layer with 64 kernels, followed by two more layers with 3x3, 64 kernels, and 1x1, 256 kernels. These three layers are iterated a total of three times.The procedure was repeated up to the fifth convolutional layer. Following that, average pooling is carried out using a fully connected layer, and softmax is applied for classification.

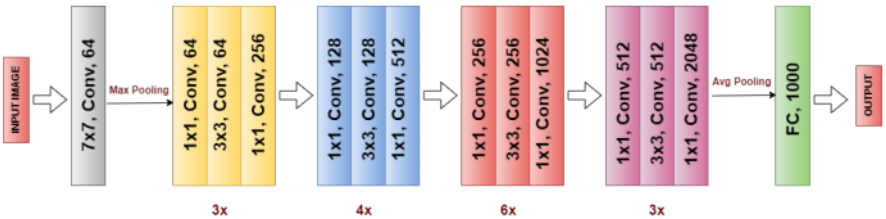


Figure 3.2: Block Diagram of ResNet50

3.1.2 MobileNetV2

MobileNetV2 [16] is a convolutional neural network-based classification model that aims to perform well on mobile devices. The main reason for adopting this architecture is its ability to reduce computational and memory costs, as well as its design's closer alignment with mobile applications.

In MobileNetV2, there are 2 types of blocks which are (i) Residual block with a stride of 1 and (ii) Residual block with a stride of 2 that is for downsizing. There are 3 convolutional layers in a block. Before the input feature map enters the depth-wise convolution layer, and 1×1 convolution layer is applied to increase the number of channels. In the middle layer, a 3 × 3 depth-wise convolution layer, filters the input feature map. After that, a 1×1 convolution layer makes up the last layer. The number of channels in the input feature map is reduced by this final convolution layer, which projects data with a large number of channels into a tensor with a much smaller number of channels. Because it limits the amount of data that can flow through the network, the last layer is also known as the bottleneck layer. Each layer has batch normalization, and ReLU is used as the activation function. But no activation function is present in the projection layer.

The MobileNet-V2 architecture has a total of 17 building blocks which are lined up in a row. These building blocks are connected by residual connections. Then the next layers are a regular 1×1 convolution, a global average pooling, and finally a classification layer.

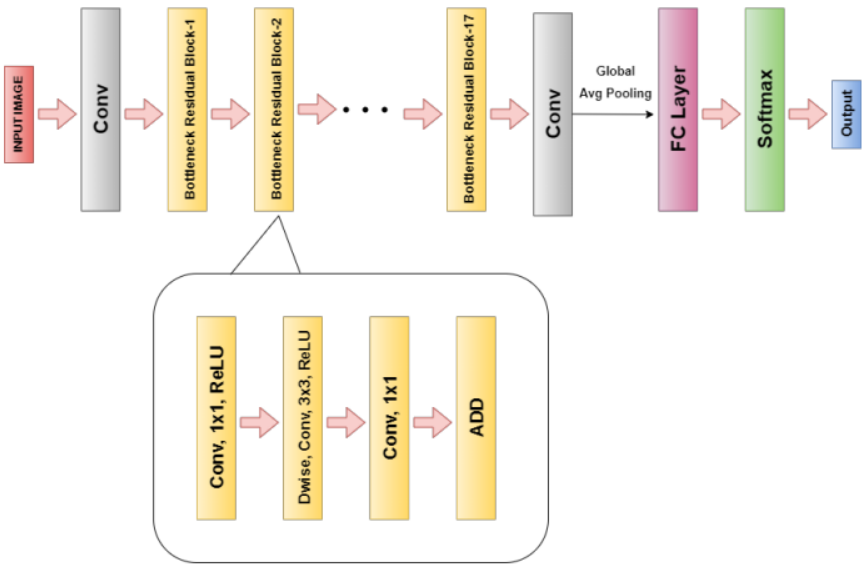


Figure 3.3: Block Diagram of MobileNetV2

3.1.3 DenseNet169

Dense convolutional networks (DenseNet) developed by Huang et al. [17] had the best classification performance on the ImageNet and CIFAR-10 datasets. Instead of using direct connections between the network's hidden layers, as in the ResNet architecture, the DenseNet architecture makes use of dense connections to build the model. There are many different variations of the DenseNet architecture. Densenet169 effectively addresses the vanishing gradient problem while having 169 layers and fewer parameters than other models.

All layers in DenseNet-169 are densely connected to one another directly using Dense Blocks. There are N(N+1)/2 direct connections, where N represents the layers number. To keep the system feed-forward, each layer receives extra inputs from all earlier layers and transmits to all subsequent layers its own feature maps. As a result, Important features are therefore shared among all layers of the network as they are learned. It also eliminates the need to learn redundant information, resulting in a significant reduction in the number of parameters.

DenseNet-169 consists of four DenseBlocks. These dense blocks are followed by a 1x1 Convolution layer and 2x2 average pooling layer. There is a 7x7 stride size of 2 Conv Layer which is followed by a 3x3 stride 2 MaxPooling layer in the first part of the architecture of DenseNet169. After that, there is a Classification Layer that performs classification by utilizing the feature maps of all of the DenseNet169 network layers.

Throughout the architecture, the model employs the ReLU activation function. Global average pooling is performed at the end of the dense block. Finally, softmax is implemented using a single fully connected layer.

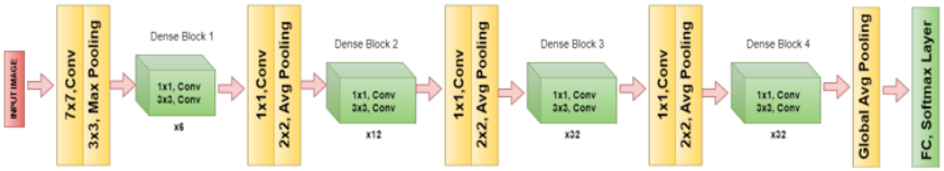


Figure 3.4: Block Diagram of DenseNet169

3.1.4 VGG16

Pre-trained models typically aid in better initialization and convergence when the dataset is relatively small and this outcome has also been widely applied in other medical imaging fields. As a result of this, we employed CNN based VGG (Visual Geometry Group) network, proposed by Simonyan et al. [18], which is regarded as one of the best computer vision model architectures. It is currently the most preferred algorithm for categorizing images and is simple to use with transfer learning.

Although many subsequent works enhanced VGG architecture. Among all configurations, the VGG16 configuration was chosen as the top-performing model. It has been trained on the ImageNet dataset, which contains millions of images from different classes.

The VGG16 has thirteen convolutional layers as well as three fully connected layers. These convolutional layers are divided into five blocks, and max pooling is done in each of the blocks. In the Input Layer, the VGG16 accepts an image input size of 224×224 RGB images. The input image is processed by a series of convolutional layers. In order to maintain the same spatial resolution between each activation map as the preceding layer, convolutional layers employ 3×3 kernels with padding sizes of 1 and strides of size 1. A 2×2 pixel window is used for the max-pooling process, with a stride length of 2. In case of the number of filters, it has 64 filters in the first layer. Then in the next layers, it has respectively 128 and 256 filters, and in the last layer, there are 512 filters .To reduce the spatial dimension, rectified linear unit (ReLU) activation is applied immediately after each convolution, and a max pooling operation is performed at the end of the each block.

Lastly, three fully connected layers are used, with a flattening layer in between following the stacks of convolutional layers. The final fully connected layer serves as the output layer and has 1000 neurons, which correspond to each of the first two's 4096 possible classes in the ImageNet dataset. And later, the Softmax activation layer is used for classification.

VGGNet has a total of around 138 million parameters. The parameters are updated for improved accuracy, and the parameter values can be used once the model has been pre-trained on a dataset. However, VGGNet16 architecture's ease is what makes the network more appealing. Just by looking at its architecture, it can easily be said that it is definitely uniform.

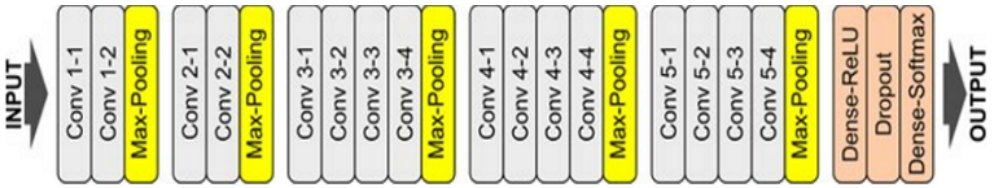


Figure 3.5: Block Diagram of VGG16

3.1.5 VGG19

In the 2014 ILSVRC competition, Simonyan et al. [17] introduced the VGG network which finished runner-up in the localization and classification tracks. VGG-19, a variation of VGG architectures, has the same basic architecture as VGG16, with three additional convolutional layers and consistently outperforms other models in terms of performance. The model is made up of highly connected convolutional and fully connected layers, allowing for more accurate feature extraction.There are 5 blocks of 16 convolution layers in VGG-19. The Input Layer of the VGG19, accepts RGB images with a size of 224x224.The Maxpooling layer, which follows each block, reduces the input image's size by two and raises the convolution layer's number of filters by two. With stride 1, it only employs 3×3 filters which are followed by numerous non-linearity layers to improve the model's ability. This boosts the network's depth and aids in learning more intricate features. The impressive VGG19 results demonstrated that a high level of classification accuracy can be attained by considering the network depth.

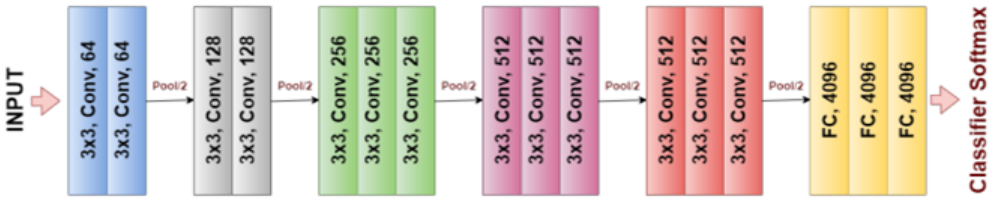


Figure 3.6: Block Diagram of VGG19

3.2 Cost-Sensitive Learning

Cost- Sensitive learning is an area of learning where the costs associated with class-imbalanced data are taken into account. This type of learning aims at minimizing the total costs and achieving maximum precision in the classification of data within a set of known classes. Cost-sensitive Learning is an approach to dealing with class imbalanced data problem. It sets out the costs of misclassification for each class and, in a way, it determines which classes are to be assigned certain weights based on their cost. Depending on the application, there are different ways to create the cost matrix. For example, in medical diagnosis, misclassifying a serious disease as benign can have serious consequences, whereas misclassifying a benign disease as serious may not have as severe consequences. In this case, the cost of misclassifying a serious disease as benign should be higher than the cost of misclassifying a benign disease as serious.

Another way to create the cost matrix is to put class weights according to the distribution of class labels. Equation 3.1 is a default log loss function which, if used in classification tasks,

puts equal weights to all the classes despite the class distribution, that leads to biasness in unbalanced data.

$$LogLoss = 1/N \sum_{i=1}^{N} [-(y_i * log(\bar{y}_i) + (1 - y_i) * log(1 - \bar{y}_i))] \tag{3.1}$$

Therefore, if used Equation 3.2, it puts proper weights to all the classes according to the class distribution in the unbalanced data.

$$WeightedLogLoss = 1/N \sum_{i=1}^{N} [-(w_0(y_i * log(\bar{y}_i)) + w_1((1 - y_i) * log(1 - \bar{y}_i)))] \tag{3.2}$$

Several studies have investigated the effectiveness of cost-sensitive learning in image classification.For example, in a study by S. H. Khan et al. [19], a cost-sensitive deep neural network was developed for multi-class image classification. The model was trained on the CIFAR-10 dataset using a cost matrix that assigned higher weights to misclassifications between classes that were more similar. The results showed that the cost-sensitive model outperformed traditional classification algorithms in terms of accuracy and misclassification costs. Vinayakumar Ravi1 et al. [20] experimented a cost-sensitive strategy to handle the negative effects of a pediatric pneumonia imbalance Database of CXR images. The experimental finding indicates that, when compared to the other cost-sensitive models, the cost-sensitive models performed better in both the macro and weighted metrics.

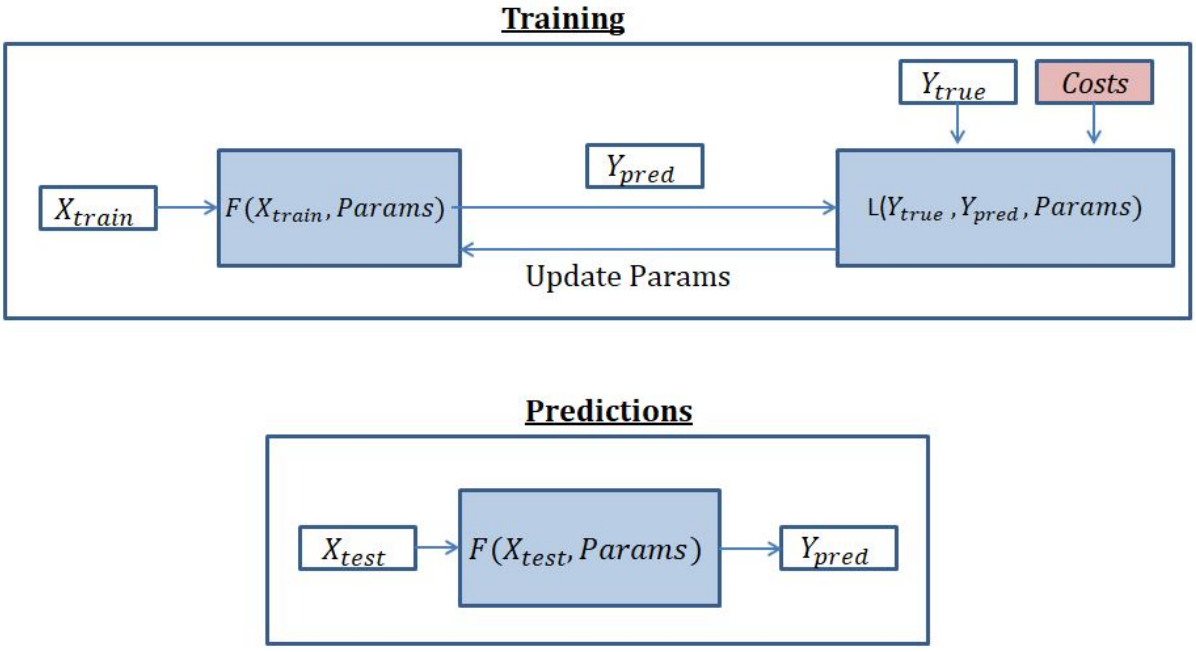


Figure 3.7: Cost-Sensitive Learning

3.3 Supervised Contrastive Learning

Supervised Contrastive learning [9] is a machine learning technique used to learn the general features of a dataset by teaching the model which data points are similar or different. The goal is to maximize the similarity between positive pairs((images from the same class) and minimize the similarity between negative pairs (images from a different class).

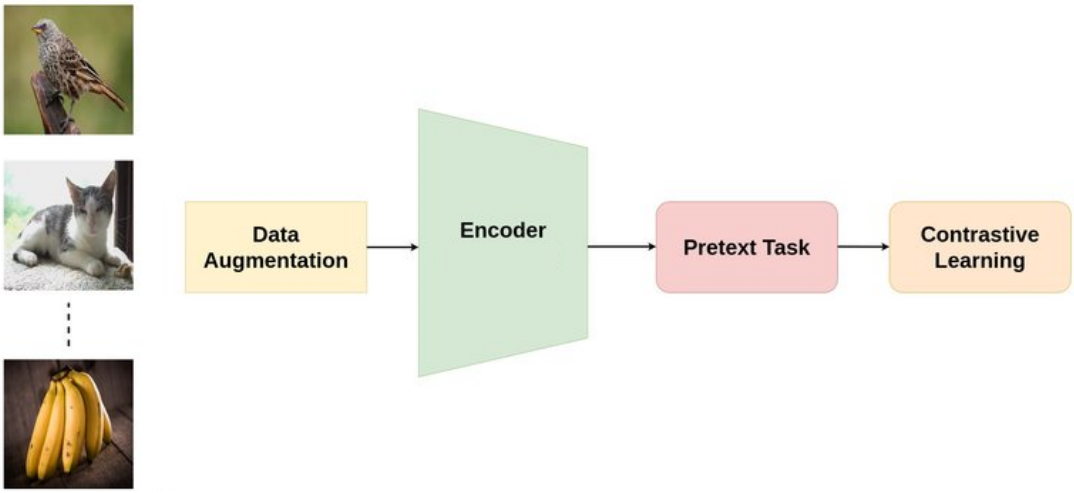


Figure 3.8: Supervised Contrastive Learning

The model can capture useful patterns and representations that can generalize well to downstream tasks by learning from this large amount of data. Supervised Contrastive Learning is a training methodology that outperforms supervised training with cross-entropy on classification tasks. In supervised contrastive learning, the training process consists of two phases-

- **Training an encoder:** In this phase, an encoder network is trained to produce vector representations (embeddings) of input images. The objective is to ensure that the embeddings of images belonging to the same class are closer to each other in the embedding space compared to embeddings of images from different classes. This is achieved by formulating a contrastive loss function that encourages the model to learn discriminative embeddings. The contrastive loss can be based on similarity measures such as cosine similarity or Euclidean distance.
- **Training a classifier on top of the frozen encoder:** After training the encoder, the embeddings are frozen, and a classifier is trained on top of these fixed embeddings. The classifier takes the embeddings as input and learns to predict the corresponding class labels. This classification training is typically performed using cross-entropy loss or any other suitable loss function for the specific classification task.

In our implementation of supervised contrastive learning, we have utilized three different loss functions.

- **Triplet Loss:** Triplet loss [21] operates on a triplet of vectors whose labels follow $y_i = y_j$ and $y_i \neq y_j$. That is to say, two of the three vectors, z_i and z_j, share the same label, while the third vector z_k has a different label. In the triplet learning literature, these vectors are termed the anchor (z_i), positive (z_j), and negative (z_k), respectively. Triplet loss is defined as:

$$L(z_i, z_j, z_k) = max(0, ||z_i - z_j||_2^2) - ||z_i - z_k||_2^2 + m) \tag{3.3}$$

- **Multi-class N-pair loss:** The multi-class N-pair loss [22] is a generalization of triplet loss, allowing for joint comparison among more than one negative sample. When applied to a pair of positive samples z_i and z_j that share the same label ($y_i = y_j$) from a mini-batch with $2N$ samples, it is calculated as:

$$L(z_i, z_j) = log(1 + \sum_{k=1}^{2N} exp(z_i z_k - z_i z_j) \tag{3.4}$$

- **Supervised NT-Xent loss:** Supervised NT-Xent [9] loss is a modification of the multi-class N-pair loss with the addition of the temperature parameter (τ) to scale the cosine similarities.

$$L(z_i, z_j) = -log \frac{exp(z_i z_j / \tau)}{\sum_{k=1}^{2N} exp(z_i z_j / \tau)} \tag{3.5}$$

3.4 Explainable Artificial Intelligence (XAI)

Explainable Artificial Intelligence (XAI) is a field of research that focuses on making AI models more interpretable and transparent to humans. One of the primary goals of XAI is to provide explanations for the decisions made by AI models. This is particularly important in cases where the decisions have significant consequences, such as in healthcare. It can be a crucial step for enhancing transparency and trust in AI systems. By providing understandable explanations for AI predictions and decisions, XAI can enable users to comprehend and validate the reasoning behind the outcomes.
There are 2 common kind techniques used in XAI for model interpretation, visualization, and feature attribution.

1. **Gradient-Based XAI:** Gradient-based XAI refers to a class of explainability techniques that utilize gradients to understand the contribution of input features towards the model's predictions. These techniques involve computing gradients with respect to the input data to identify the features that have the most significant influence on the model's output. Some XAI methods that are based on Gradients are GradCAM [23], GradCAM++ [24], ScoreCAM [25], and LayerCAM [26]. GradCAM or Gradient Weighted Activation Mapping Technique, is a class-discriminative localization technique. It computes the importance score based only on the gradients flowing into the final convolutional layer of the CNN. GradCAM++ differs from GradCAM in the computation method, as it takes into account both the forward and backward gradients in its computation. ScoreCAM, on the other hand, considers both the forward and backward gradients but uses a global average pooling operation to obtain a single importance score for each feature map of the final convolutional layer, which is then used to compute a weighted combination of the feature maps to produce the final heatmap. Another XAI technique that generates class activation maps by analyzing various layers of a deep neural network is LayerCAM. By examining the activations at different layers, LayerCAM identifies the regions in an input image that contribute significantly to the model's classification decision. Figure 3.9 represents how different gradient-based XAI methods such as GradCAM, GradCAM++, a faster version of ScoreCAM and LayerCAM generates heatmaps and interpret classifier's decision-making process.

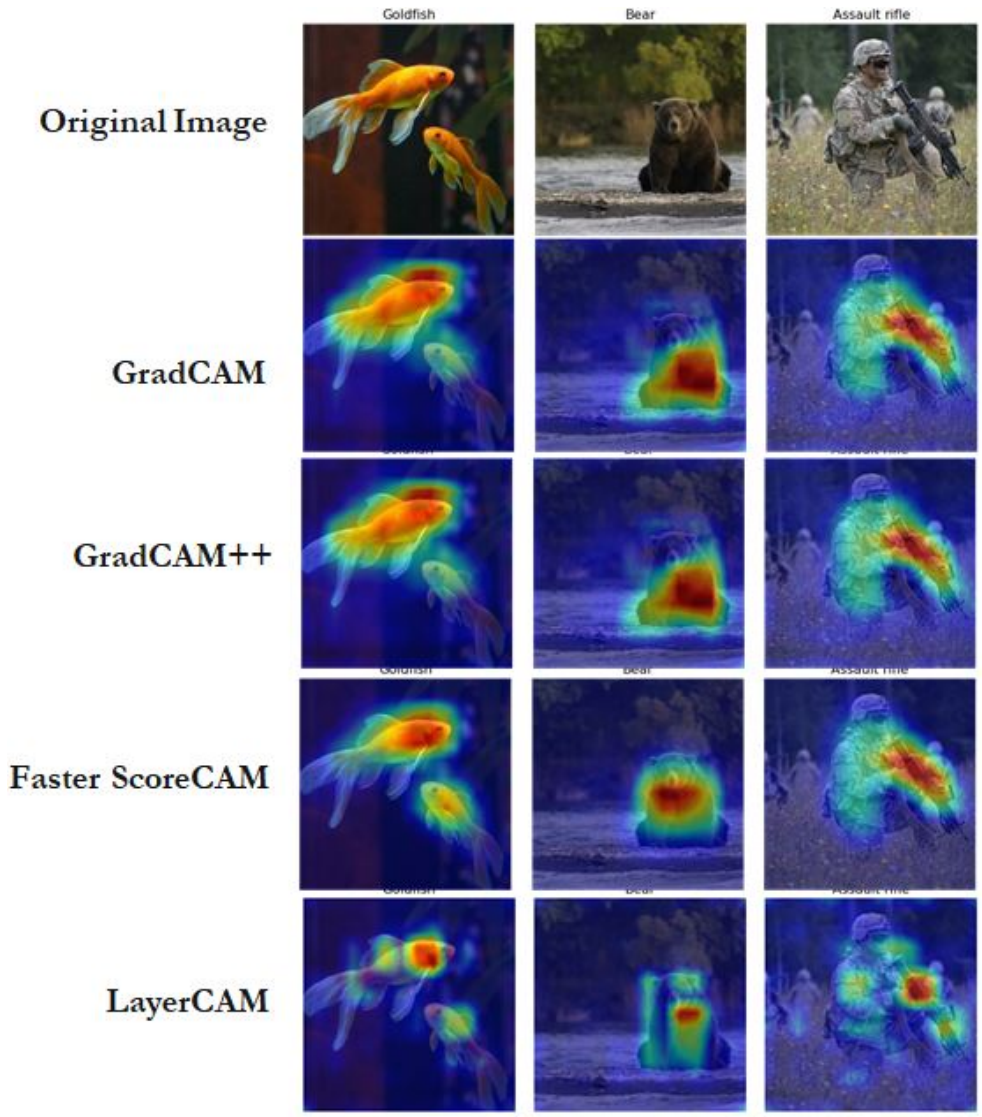


Figure 3.9: Visualisation of three different samples using gradient-based XAI techniques[1]

2. **Perturbation-Based XAI:** Perturbation-based XAI methods are techniques that involve perturbing or modifying the input data to observe the impact on the model's predictions. These methods aim to understand the sensitivity of the model's output to changes in the input features. A method that is based on perturbation is LIME or Local Interpretable Model Agnostic Explanation [27]. It provides local explanations by generating random perturbations to determine the significance of each feature in the classification. It is model-agnostic and provides explanations without making any assumptions about the model. Figure 3.10 shows a sample image, the image with superpixels that contribute positively to the class and finally the image with both pros and cons of the prediction.

[1] https://github.com/keisen/tf-keras-vis

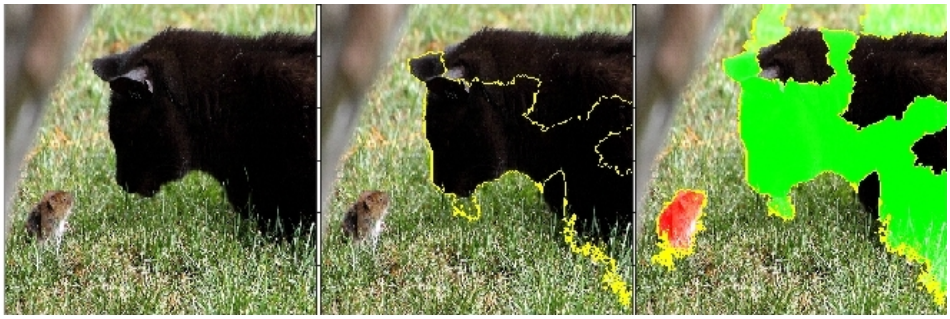


Figure 3.10: Visualisation of a sample image using Perturbation-Based XAI[2]

By generating random perturbations of the input data, LIME identifies the top superpixels that contribute positively to the predicted class. These important regions are highlighted while the rest of the image remains present. Additionally, LIME uses color-coded visualizations to showcase the "pros and cons" of the prediction, with pros represented in green and cons in red. This approach enables interpretable insights into the model's decision-making process and helps us understand the factors influencing the model's output.

[2] https://github.com/marcotcr/lime/tree/master/doc/notebooks

Chapter 4

Dataset Description

Chapter 4 focuses on the data statistics and data split for the conducted study. The "Data Statistics" section describes the dataset, including its origin, collection methodology, and notable characteristics. It also provides statistical information such as dataset size, class distribution, number of images in each class. In the "Data Split" section, the process of splitting the dataset into train, validation, and test sets is explained, along with the augmentation techniques employed. It also includes the number of images in each subset, offering insights into the dataset's distribution.

4.1 Data Statistics

The SIPaKMeD Dataset [8] is a dataset of pap smear images, where medical experts have manually defined the nucleus and the area of cytoplasm in each image and have labeled the images with utmost care and expertise.
The SIPaKMeD Dataset includes images that have been cropped manually from the images of the Pap Smear test. The Pep smear, which is also known as the Papanicolaou test, is one of the most popular screening tests for cervical cancer. It is a test that can show anomalies in the cervical cells that can later develop into cancer. The SIPaKMeD dataset includes 4049 precisely cropped photos of isolated cells and 966 images of Pap smear slide cluster cells. These images were captured with an optical microscope CCD camera (OLYMPUS BX53F) [8].
The dataset consists of images of five types of cancer cells that are-

(a) **Superficial Intermediate:** Most of the cells found in the Pap smear test are superficial intermediate cells. They are usually oval, polygonal, or round in shape. The superficial intermediate cells are known to be normal cells that are not cancerous.

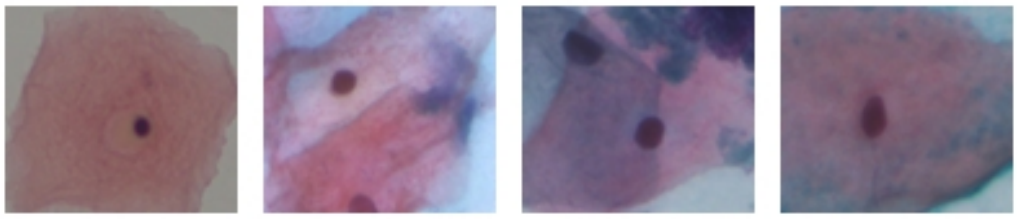


Figure 4.1: Superficial Intermediate Cells

(b) **Parabasal:** Another type of normal cell that is not cancerous is the parabasal cell. They are very small in size. They are generally immature Squamous cells. As they have morphological similarities with Metaplastic cells, sometimes differentiating between them can be hard.

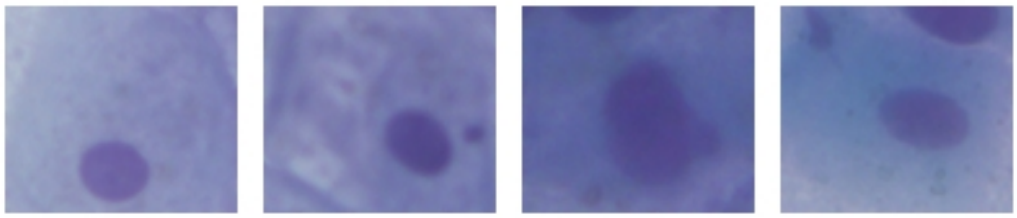


Figure 4.2: Parabasal Cells

(c) **Koilocytotic:** Koilocytic cells are pathognomonic for HPV infection, and depending on the stage of infection and the virus type that has been contracted, their nuclei frequently show varying degrees of degeneration.

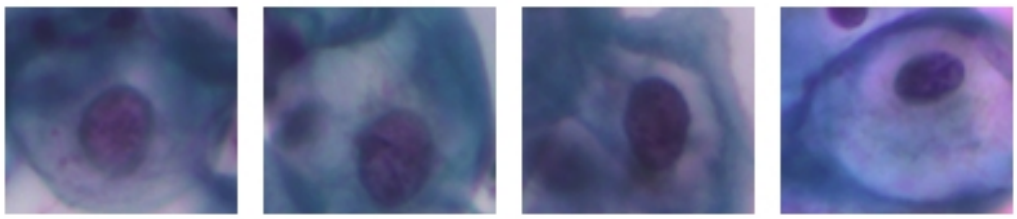


Figure 4.3: Koilocytotic Cells

(d) **Dyskeratotic:** Squamous cells that have experienced premature aberrant keratinization, either individually or more frequently in three-dimensional clusters, are referred to as dyskeratotic cells. They show noticeable signs of HPV infection.

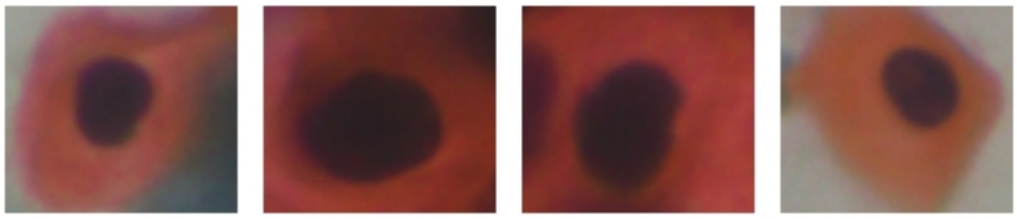


Figure 4.4: Dyskeratotic Cells

(e) **Metaplastic Cells:** Metaplastic cells have morphological similarities with Parabasal cells. But because of the unusual, almost spherical shape of their cytoplasm, greater pre-cancerous lesion detection rates are associated with their appearance in the Pap test.

Figure 4.5: Metaplastic Cells

Table 4.1 presents the distribution of the five types of cells in the Sipakmed dataset-

Table 4.1: Distribution of Images in SIPaKMeD dataset

Category	**Number of Image**
Superficial-Intermediate	813
Parabasal	727
Koilocytotic	825
Metaplastic	793
Dyskeratotic	813

4.2 Data Split

The SIPaKMeD dataset consists of 4049 images. These images were split into three sets: train, test, and validation. The train set comprises 80% of the total images. The test set contains 20% of the images, resulting in 812 test images. The train set was further divided into train and validation subsets, with each subset having 80% and 20% of the train images, respectively. Therefore, the number of train images is 2589, the number of validation images is 648 and the number of test images is 812.

After splitting the original train set, we applied various augmentation techniques to introduce more diversity into the training data. These techniques included Affine transformations such as random rotation, translation, scaling, shearing, zooming, flipping, and padding, which artificially generated new variations of the input data. We also incorporated noise injection, contrast adjustment, brightness modification, and changes in specific pixel values to further increase the variability in the training samples. As a result, the augmented train set expanded to a total of 18123 images, providing a richer and more diverse dataset for training the model.

Table 4.2 represents the summary of dataset partition and total number of images after augmentation.

Table 4.2: Summary of Dataset Partition

Set	**Number of Images**
Train	18123
Validation	648
Test	812

Chapter 5

Methodology

This chapter represents the methodology employed in the study, focusing on five key subsections. The "Input" subsection describes the dataset used, highlighting its size and specific characteristics. "Image Preprocessing" details the steps taken to enhance the input data's quality and suitability. The "Training" subsection discusses the classifiers, their architecture, and experiments for model training. "Performance Evaluation" covers the metrics and methodologies employed to assess the model's performance. Lastly, the "Interpretation" subsection explores techniques used to interpret the model's results and gain insights. Together, this chapter provides a comprehensive overview of the research methodology, guiding through the input, preprocessing, training, evaluation, and interpretation stages.

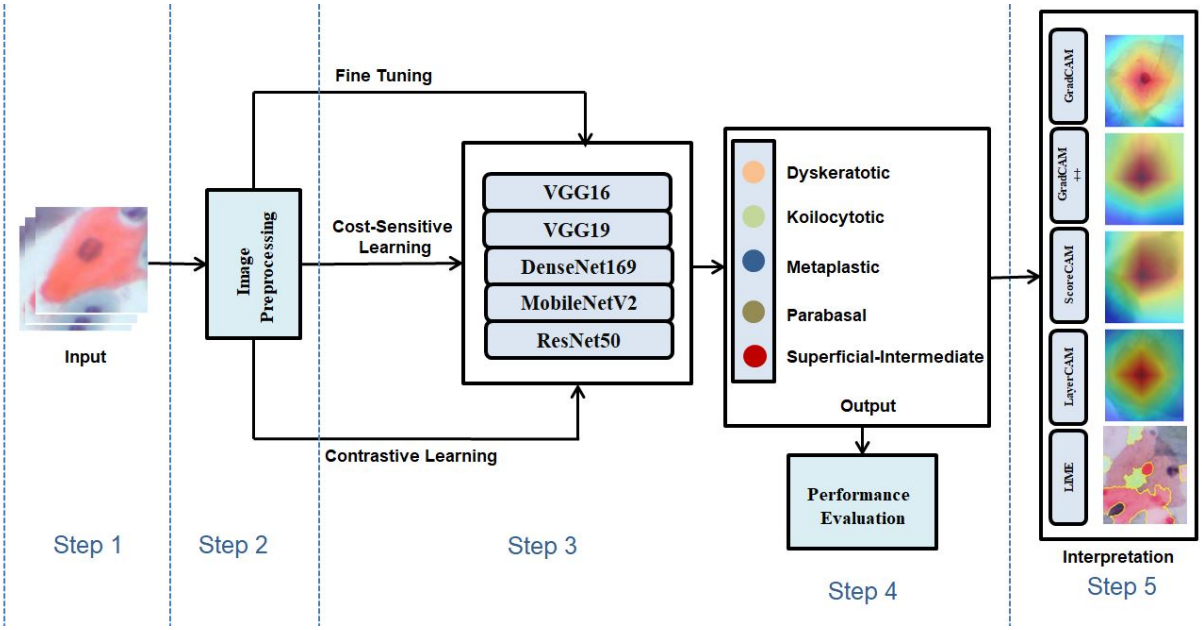


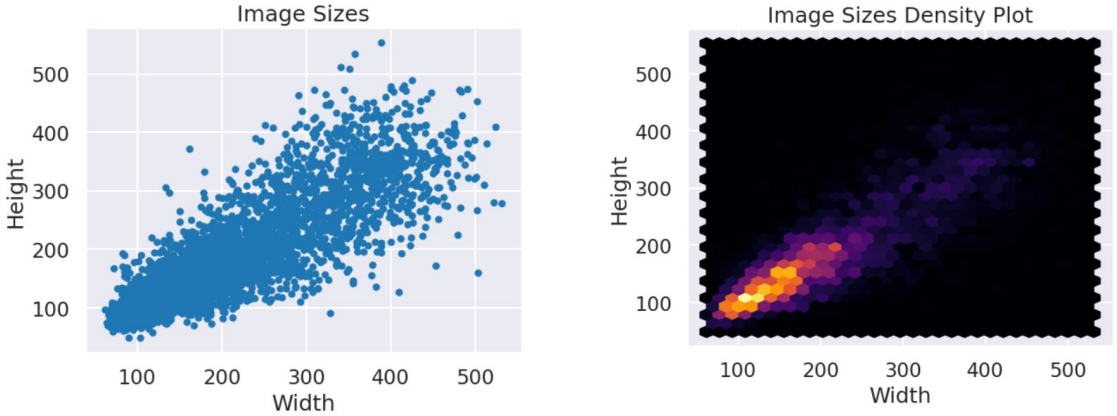
Figure 5.1: Methodology

5.1 Step 1- Input

As discussed in Chapter 4, we used the SIPaKMeD Dataset images for our experiments, images of cervical cells that have been cropped manually from the images of the Pap Smear test.

5.2 Step 2- Image Preprocessing

To ensure consistency and compatibility across our dataset, we decided to perform a standardization step by resizing all images to a uniform resolution. The minimum resolution of the images in the dataset is 62x48 pixels, whereas the maximum resolution of the images in the dataset is 531x553 pixels. Downscaling larger images makes it more difficult for CNN to learn the crucial features needed for classification or detection because the number of pixels where the feature would be present is drastically decreased, whereas small images that have been upscaled and zero-padded require NN to learn that the zero-padded component has no importance on classification. Larger images may need more VRAM and take longer to train. To find out the optimal resolution for our images, we analyze all the images by creating a scatter plot using matplotlib, where the x-axis represents the widths and the y-axis represents the heights of the images. Each point on the plot represents an image, and the resulting plot visualizes the distribution of image sizes. We also created a 2D density plot using matplotlib's hexbin function, visualizing the distribution of image sizes. The resulting plot shows the density of image sizes based on width and height.



(a) Scatter Plot

(b) Density Plot

Figure 5.2: A Distribution-Based Analysis for Image Size Determination

Finally, We calculated the height and width of each image. We created a 2D histogram of the heights and widths, and identified the bin with the highest count, representing the most common size, which is 98.5x108.9 pixels. For easy computation, we chose 110x110 pixels

and resized all the images to this optimal size. This resizing process allowed us to maintain the aspect ratio of the images while ensuring that they were all of the same dimensions, regardless of their original size. By doing so, we made sure that the images were ready for further processing and analysis, as they now had a consistent format that could be used by our machine learning algorithms. Reshaping the images to this specific resolution allowed us to proceed with subsequent steps of our workflow with greater efficiency and accuracy, ultimately leading to improved performance and better results.

In order to improve the performance and generalization of our machine learning model, we implemented several data augmentation techniques. These techniques were designed to introduce more diversity into the training data and enhance the model's ability to recognize and classify different patterns and features. We utilized methods such as Affine transformation that includes random rotation, translation, scaling, shearing, zooming, flipping, and padding to artificially generate new variations of the input data. Additionally, we applied noise injection, contrast adjustment, brightness modification, and changes in specific pixel values to further increase the variability in the training samples. Through these augmentation strategies, we aimed to enable the model to better handle real-world scenarios and generalize to new, unseen data with higher accuracy and reliability.

5.3 Step 3 - Training

(a) **Fine-Tuning:** We performed fine-tuning on five pre-trained deep learning models: VGG16, VGG19, DenseNet169, MobileNetV2, and ResNet50. We used the pre-trained weights of these models, which had already been trained on large datasets, to start our models with effective weights. We then fine-tuned these models on our specific cervical cell image dataset by retraining the last few layers of the network, while freezing the weights of the earlier layers.

For VGG16, we freeze the first 13 layers, adding dropout layer and BatchNormalization for regularization, and replaced the top layers with new fully connected layers to classify the images into five classes. Similarly, for VGG19, we freeze the first 17 layers, adding dropout layer and BatchNormalization for regularization, and replaced the top layers with new fully connected layers for the classification task.

In DenseNet169, we freeze the first 249 layers, allowing only the layers starting from the 249th layer to be updated during training. In MobileNetV2, we freeze the first 100 layers and allowed updates in the layers starting from the 100th layer. In ResNet50, we freeze the first 86 layers and trained the layers from the 86th layer onwards.

We were able to refine the models for our specific classification task through the process of fine-tuning with selective layer freezing. Because of this, we were able to

achieve highly accurate results proving the value of transfer learning for image classification in our study.

(b) **Cost-Sensitive Learning:**

As our dataset is slightly imbalanced, we applied Cost-Sensitive Learning, and made the 5 classifiers cost sensitive by assigning different costs to our classes. We have assigned 0.996319018404908, 0.9842424242424243, 1.0213836477987421, 1.0278481012658227, and 0.9724550898203593 cost respectively to Dyskeratotic, Koilocytotic, Metaplastic, Parabasal, and Superficial-Intermediate class. Then we trained the classifiers with the costs assigned to each class.

(c) **Supervised Contrastive Learning:**

We applied supervised contrastive learning along with n pairs contrastive loss function to train our five different classifiers: VGG16, VGG19, DenseNet169, MobileNetV2, and ResNet50. Contrastive learning is a self-supervised learning technique that aims to learn useful representations by maximizing the agreement between similar instances and minimizing it for dissimilar instances. The contrastive loss function is designed to pull together similar instances in the feature space while pushing apart dissimilar instances. The training process involves creating pairs of samples, where each pair consists of two images. For each pair, a similarity label is assigned, indicating whether the images are from the same class (positive pair) or different classes (negative pair). The contrastive loss is then computed based on the feature representations of these pairs, encouraging the model to learn discriminative features.

In our study, the supervised contrastive learning involves a series of steps to extract meaningful representations from input images. The process can be summarized as follows:

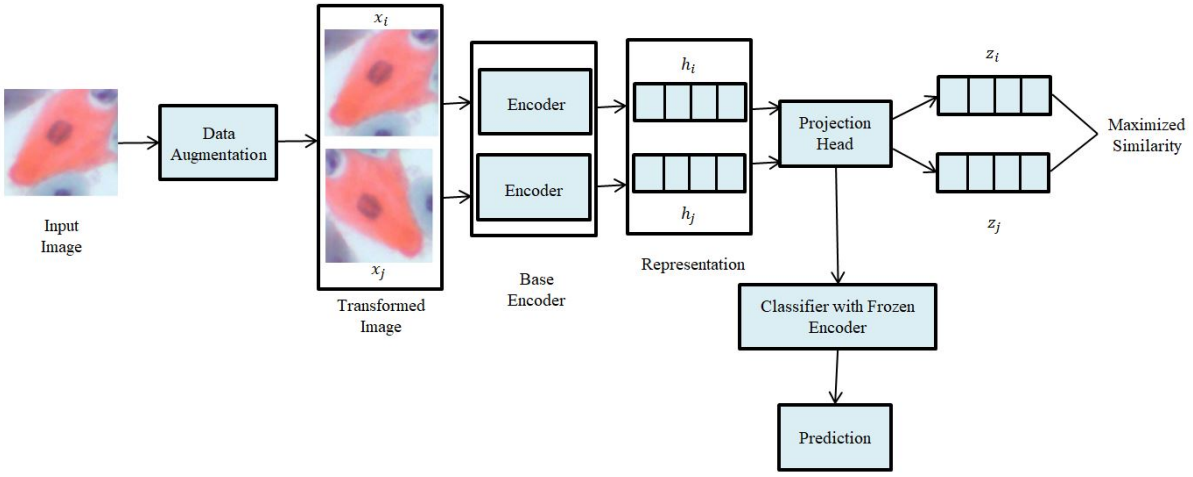


Figure 5.3: Steps of Supervised Contrastive Learning

First, the input image undergoes data augmentation. This involves normalization, random horizontal flipping, random rotation, random width scaling, and random height scaling to create pairs of augmented samples. These pairs consist of an anchor image and a positive image that share the same class label.

The augmented samples are then passed through a base encoder, which is our deep neural networks. The base encoder extracts high-level features from the input images and maps them into a continuous representation space.

The representations obtained from the base encoder are then fed into a projection head. The projection head consists of fully connected layers that project the representations into a different space. This projection step helps to maximize the similarity between the representations of positive pairs while minimizing the similarity between representations of negative pairs.

After the projection, the representations are passed through a frozen encoder. The frozen encoder is a copy of the base encoder with its weights fixed. It serves as a feature extractor, and its purpose is to capture additional information from the representations. Finally, the frozen encoder's output is used to make predictions on the labeled data. The predictions are then compared to the ground truth labels to compute a loss and update the model's parameters.

By applying contrastive learning and the contrastive loss to the classifiers, the models benefit from improved feature representations, leading to better performance.

5.4 Step 4- Performance Evaluation

After training the five classifiers (VGG16, VGG19, DenseNet169, MobileNetV2, and ResNet50), we evaluated their performance using standard evaluation metrics: precision, recall, F1 score, and accuracy. Precision is the ratio of correctly predicted positive observations to the total predicted positive observations [28]. Recall is the ratio of correctly predicted positive observations to the all positive observations in the actual class [29]. F1 Score is the weighted average of Precision and Recall. Accuracy indicates rate of correct classification. It is simply a ratio of correctly predicted observation (true positive and true negative) to the total number of observations [30].

By assessing these metrics, we measured the classifiers' ability to correctly classify images, considering both false positives and false negatives. These evaluation metrics offer a comprehensive understanding of the classifiers' performance, indicating their precision, ability to identify positive instances, overall correctness, and trade-off between precision and recall. Analyzing the precision, recall, F1 score, and accuracy of the trained classifiers allowed us to assess their effectiveness and select the model that achieves the desired balance between

these metrics for a given application. It provides insights into the classifiers' performance in handling different classes, understanding their strengths and weaknesses, and guiding further improvements or model selection based on specific requirements and objectives.

5.5 Step 5- Interpretation

We used 2 kinds of techniques to explain the decision making of our classifiers. (i) Gradient-Based Methods, and (ii) Perturbation-Based Methods.
Methods we used that are based on Gradients are GradCAM [23], GradCAM++ [24], ScoreCAM [25], and Faster ScoreCAM [31]. GradCAM or Gradient Weighted Activation Mapping Technique, is a class-discriminative localization technique. It computes the importance score based only on the gradients flowing into the final convolutional layer of the CNN. GradCAM++ differs from GradCAM in the computation method, as it takes into account both the forward and backward gradients in its computation. ScoreCAM, on the other hand considers both the forward and backward gradients but uses a global average pooling operation to obtain a single importance score for each feature map of the final convolutional layer, which is then used to compute a weighted combination of the feature maps to produce the final heatmap. Another method we used that is based on perturbation is LIME or Local Interpretable Model Agnostic Explanation [27]. It provides local explanations by generating random perturbations to determine the significance of each feature in the classification. It is model-agnostic and provides explanations without making any assumptions about the model. All of the techniques provided visual explaination of how our classifiers came to such decision to classify the cervical cell images.

Chapter 6

Evaluation

6.1 Evaluation Metrics

In any classification models, it is very important to evaluate the models to determine how good it performs. To evaluate our models, we have used a variety of evaluation metrics such as weighted accuracy, weighted precision, weighted recall, and weighted F1 score.

Accuracy: Accuracy indicates rate of correct classification. It is simply a ratio of correctly predicted observation (true positive and true negative) to the total number of observations [30]. Hence,

$$Accuracy = \frac{\text{Correctly Predicted Observations}}{\text{Total Observations}} \tag{6.1}$$

Precision: Precision is the ratio of correctly predicted positive observations to the total predicted positive observations [28].

$$Precision = \frac{\text{True Positive Observations}}{\text{Total Predicted Positive Observations}} \tag{6.2}$$

Recall: Recall is the ratio of correctly predicted positive observations to the all positive observations in the actual class [29].

$$Recall = \frac{\text{True Positive Observations}}{\text{Total Positive Observations in the Class}} \tag{6.3}$$

F1 Score: F1 Score is the weighted average of Precision and Recall.

$$F1Score = \frac{\text{2*Recall*Precision}}{\text{Recall+Precision}} \tag{6.4}$$

The weighted metrics are measured by taking into consideration the occurrences of the classes in the dataset.

6.2 Hyper-parameter Settings

The experiments to evaluate our methodology on our chosen dataset have been performed on Google Colaboratory and Kaggle notebook. Google Colaboratory and Kaggle are two online platforms that allow to work collaboratively. Google Colab is a free, virtual machine environment that offers pre-configured data science libraries such as Pandas and Scikit-learn, and also provides access to GPUs and TPUs for faster computing. Kaggle, on the other hand, is a platform that allows to create models and and provide more GPU access for machine learning and data science. Hyperparameters are parameters that are set before a model is trained that have an impact on the model's performance and behavior. These parameters are manually set rather than being learned from the data. The performance can be strongly impacted by its hyperparameters, and effective tuning can frequently improve accuracy and generalization. They hypeparameters set for our experiments are given in table 6.1.

Table 6.1: Hyperparameters Setting for Training

Experiment	Hyperparameter	Value
Fine-Tuning and Cost Sensitive Learning	Epoch	50
	Loss Function	Categorical Crossentropy
	Activation Function	Softmax
	Optimizer	Adam
	Size	110x110 pixels
	Dropout	0.5
	Learning Rate	1e-3
Contrastive Learning	Epoch	50
	Loss Function	Multiclass N-pairs Loss
	Activation Function	Softmax
	Optimizer	Adam
	Size	110x110 pixels
	Dropout	0.5
	Learning Rate	1e-3

6.3 Experimental Result

For experimental purposes, we have split our dataset by putting 80% of the images for training dataset, and 20% images for testing. Then we have split the training dataset by

putting 80% of the images for training and 20% images for validation. We have kept our batch size at 32 and shuffled the data after every epoch to prevent overfitting. We list below the results of our extensive experiments in this section with the evaluation metrics.

(i) Fine-Tuning: We have fine-tuned 5 different classifier models for our study, which are- VGG16, VGG19, DenseNet169, MobileNetV2, and ResNet50. Table 6.2 shows that, DenseNet169 achieved the best accuracy among all 5 classifiers, which is 97.17%. It also shows that ResNet and VGG19 yieled least satisfactory results, whereas VGG16 and MobileNetV2 performed somewhat better on our dataset.

Table 6.2: Performance Comparison using Fine-Tuning

Classifier	Class	Weighted Precision	Weighted Recall	Weighted F1 Score	Accuracy
VGG16	Dyskeratotic	0.94	0.96	0.95	95.57%
	Koilocytotic	0.93	0.88	0.90	
	Metaplastic	0.95	0.96	0.96	
	Parabasal	0.99	0.99	0.99	
	Superficial-Intermediate	0.97	0.99	0.98	
VGG19	Dyskeratotic	0.92	0.94	0.93	94.21%
	Koilocytotic	0.90	0.87	0.88	
	Metaplastic	0.94	0.94	0.94	
	Parabasal	0.97	0.98	0.98	
	Superficial-Intermediate	0.98	0.99	0.99	
DenseNet169	**Dyskeratotic**	**0.95**	**0.97**	**0.96**	**97.17%**
	Koilocytotic	**0.98**	**0.93**	**0.96**	
	Metaplastic	**0.95**	**0.98**	**0.97**	
	Parabasal	**0.99**	**0.99**	**0.99**	
	Superficial-Intermediate	**0.99**	**0.99**	**0.99**	
MobileNetV2	Dyskeratotic	0.98	0.96	0.97	96.55%
	Koilocytotic	0.93	0.94	0.93	
	Metaplastic	0.98	0.95	0.97	
	Parabasal	0.97	1.00	0.98	
	Superficial-Intermediate	0.98	0.98	0.98	
ResNet50	Dyskeratotic	0.94	0.96	0.95	94.58%
	Koilocytotic	0.89	0.90	0.89	
	Metaplastic	0.92	0.93	0.93	
	Parabasal	1.00	0.98	0.99	
	Superficial-Intermediate	0.99	0.96	0.98	

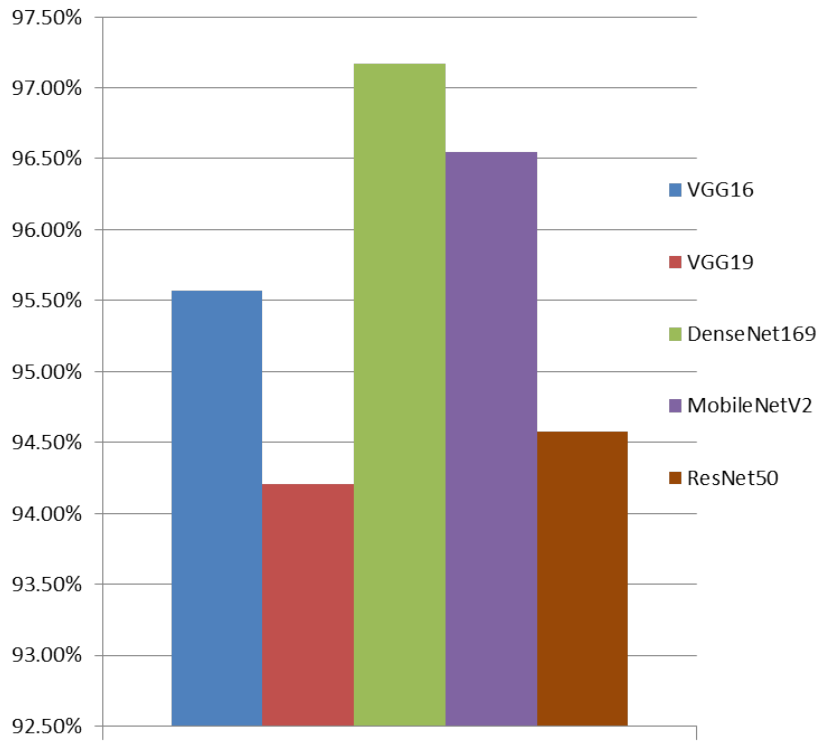


Figure 6.1: Performance Comparison with accuracy using Fine-Tuning

Figure 6.1 represents the accuracy achieved by different classifiers using fine-tuning.

(ii) Cost-sensitive Learning: We also applied Cost-Sensitive Learning, and made the 5 classifiers cost sensitive by assigning different costs to our classes. We have assigned 0.996319018404908, 0.9842424242424243, 1.0213836477987421, 1.02784810126582-27, and 0.9724550898203593 cost respectively to Dyskeratotic, Koilocytotic, Metaplastic, Parabasal, and Superficial-Intermediate class.

Table 6.3: Performance Comparison using Cost-Sensitive Learning

Classifier	Class	Weighted Precision	Weighted Recall	Weighted F1 Score	Accuracy
VGG16	Dyskeratotic	0.96	0.98	0.97	95.94%
	Koilocytotic	0.94	0.90	0.92	
	Metaplastic	0.95	0.95	0.95	
	Parabasal	0.99	0.99	0.99	
	Superficial-Intermediate	0.97	0.98	0.97	
VGG19	Dyskeratotic	0.96	0.93	0.94	94.09%
	Koilocytotic	0.88	0.90	0.89	
	Metaplastic	0.93	0.91	0.92	
	Parabasal	0.97	0.98	0.98	
	Superficial-Intermediate	0.97	0.99	0.98	
DenseNet169	**Dyskeratotic**	**0.96**	**0.96**	**0.96**	**97.29%**
	Koilocytotic	**0.96**	**0.94**	**0.95**	
	Metaplastic	**0.96**	**0.98**	**0.97**	
	Parabasal	**1.00**	**0.99**	**0.99**	
	Superficial-Intermediate	**0.99**	**0.99**	**0.99**	
MobileNetV2	Dyskeratotic	0.98	0.94	0.96	95.57%
	Koilocytotic	0.90	0.94	0.92	
	Metaplastic	0.97	0.93	0.95	
	Parabasal	0.97	1.00	0.98	
	Superficial-Intermediate	0.96	0.97	0.97	
ResNet50	Dyskeratotic	0.91	0.98	0.94	94.21%
	Koilocytotic	0.91	0.87	0.89	
	Metaplastic	0.92	0.92	0.92	
	Parabasal	0.99	0.97	0.98	
	Superficial-Intermediate	0.98	0.97	0.97	

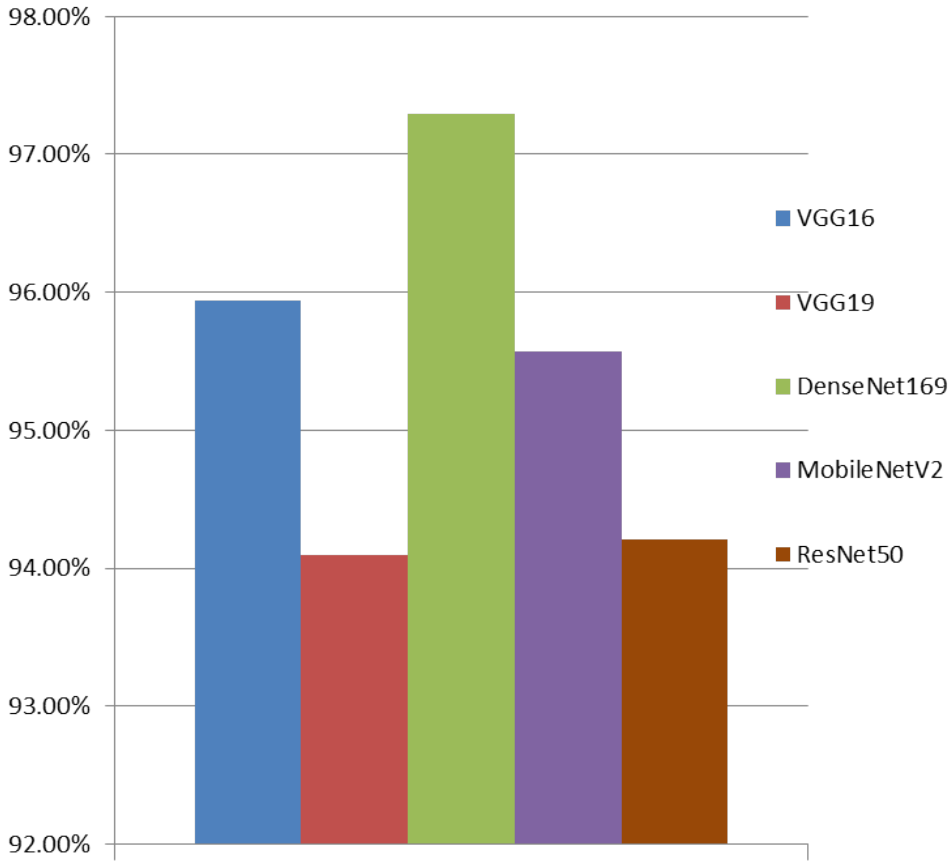
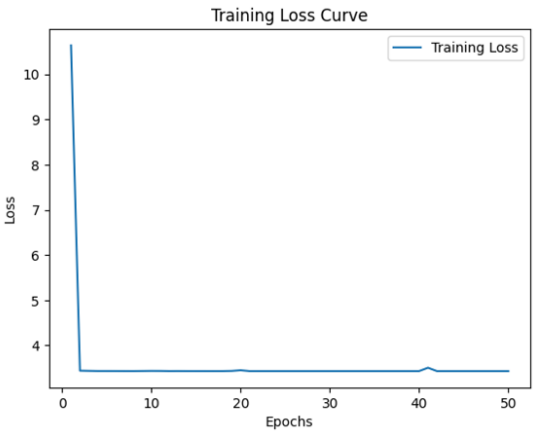


Figure 6.2: Performance Comparison with accuracy using Cost Sensitive Learning

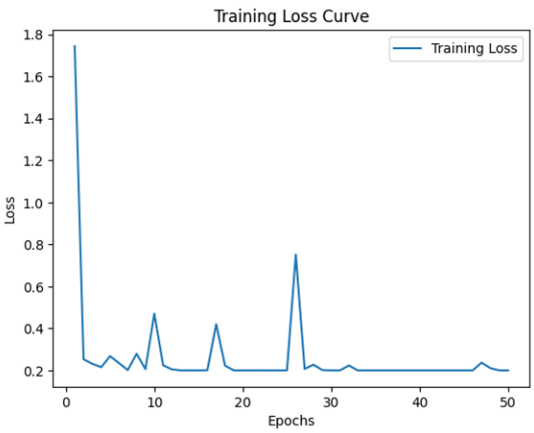
Table 6.3 and Figure 6.2 shows that, DenseNet169 achieved the best accuracy with cost sensitive learning among all 5 classifiers, which is 97.29%. It also shows that ResNet50 and VGG19 yield least satisfactory results, whereas VGG16 and MobileNetV2 performed slightly better on our dataset. It is evident from the results that, when incorporated cost sensitive learning, VGG16, and DenseNet169 performed better than fine-tuned classifiers.

(ii) Supervised Contrastive Learning: We also introduced supervised contrastive learning, incorporating contrastive learning loss with Supervised NT Xent Loss function, the Triplet Loss function and Multiclass N-pairs Loss[1]. We experimented the three loss functions with DenseNet169 model as it yield the best performance with our dataset with an accuracy of 97.29%.

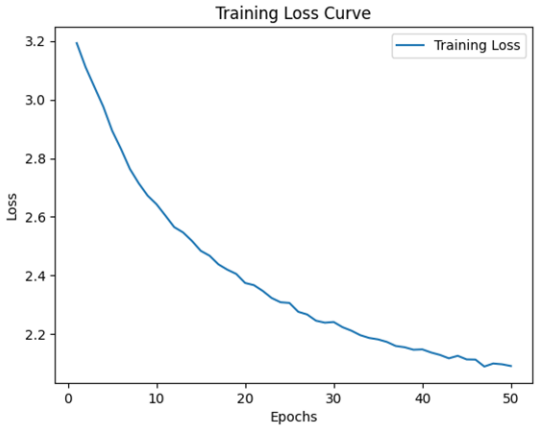
[1] https://shorturl.at/cuzGH



(a) Supervised NT Xent Loss



(b) Triplet Loss



(c) Multiclass Npairs Loss

Figure 6.3: Comparison Among Different Loss Curves

In Figure 6.3a, all 50 of the epochs on the loss curve have a consistent value of 3.4333, which indicates a lack of significant improvement in the model's performance using NT-Xent Loss function. From the first epoch, where the loss was 9.1193, the loss rapidly drops and settles at 3.4333 within the first few epochs. However, further training iterations does not lead to significant reductions in loss. In Figure 6.3b, this loss value does not significantly decrease throughout training using the triplet loss function.The loss values remain consistently high throughout training, indicating that the model has difficulty picking up new information and making accurate predictions. In Figure 6.3c, the training process demonstrates the progress of the model over 50 epochs using the multiclass n-pair Loss function. Throughout the training, we can observe a steady decline in the loss value, indicating that the model is learning and improving its ability to capture meaningful representations of the data. This decrease in loss indicates that the model is successfully maximizing similarity between samples from the same class and minimizing similarity between samples from different classes in the embedding space.

Since, Multiclass N-pairs Loss function has shown promising results, we select this loss function as the preferred loss function for contrastive learning in our experiment and apply it to all the classifiers.

Table 6.4 displays the result of supervised contrastive learning with an epoch of 50. With contastive learning, VGG16 classifier performed the best with Multiclass N-pairs Loss, which is the best performance among all the experiments conducted in our study.

Table 6.4: Performance Comparison using Supervised Contrastive Learning

Classifier	Class	Weighted Precision	Weighted Recall	Weighted F1 Score	Accuracy
VGG16	**Dyskeratotic**	**0.95**	**0.98**	**0.97**	**97.29%**
	Koilocytotic	**0.98**	**0.94**	**0.96**	
	Metaplastic	**0.96**	**0.97**	**0.97**	
	Parabasal	**1.00**	**0.99**	**0.99**	
	Superficial-Intermediate	**0.98**	**0.99**	**0.98**	
VGG19	Dyskeratotic	0.98	0.97	0.97	96.68%
	Koilocytotic	0.95	0.95	0.95	
	Metaplastic	0.94	0.96	0.95	
	Parabasal	0.99	0.97	0.98	
	Superficial-Intermediate	0.99	0.98	0.99	
DenseNet169	Dyskeratotic	0.96	0.95	0.95	93.47%
	Koilocytotic	0.90	0.86	0.88	
	Metaplastic	0.92	0.93	0.92	
	Parabasal	0.93	0.98	0.95	
	Superficial-Intermediate	0.96	0.96	0.96	
MobileNetV2	Dyskeratotic	0.95	0.95	0.95	95.81%
	Koilocytotic	0.96	0.92	0.94	
	Metaplastic	0.95	0.97	0.96	
	Parabasal	0.97	0.99	0.98	
	Superficial-Intermediate	0.98	0.96	0.97	
ResNet50	Dyskeratotic	0.95	0.96	0.95	96.43%
	Koilocytotic	0.95	0.93	0.94	
	Metaplastic	0.96	0.98	0.97	
	Parabasal	0.98	0.99	0.99	
	Superficial-Intermediate	0.98	0.96	0.97	

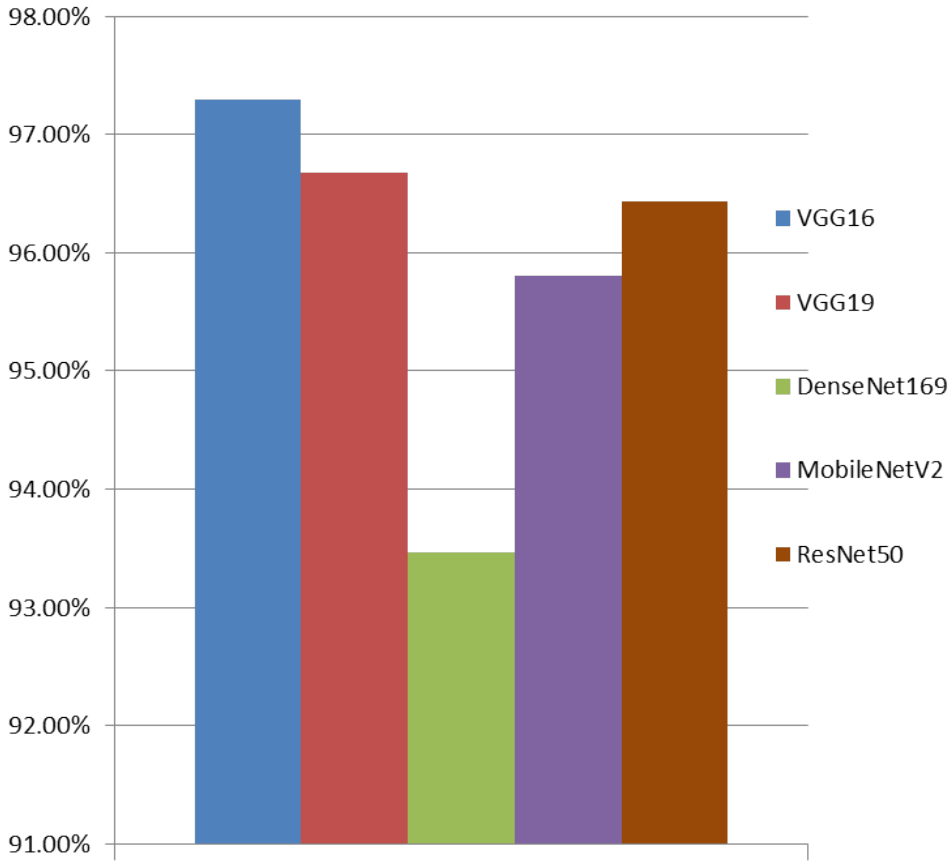


Figure 6.4: Performance Comparison with accuracy using Supervised Contrastive Learning

Figure 6.4 represents the accuracy achieved by different classifiers using supervised contrastive learning.

6.4 Interpretation using XAI

We employed a neural network visualization toolkit[2], to implement gradient-based visualization techniques such as GradCAM, GradCAM++, a faster variant of ScoreCAM, and LayerCAM. We also implemented perturbation-based visualization technique, LIME. These techniques allowed us to visualize both correct and incorrect classifications generated by our best performed classifier.

6.4.1 Gradient-based Visualization

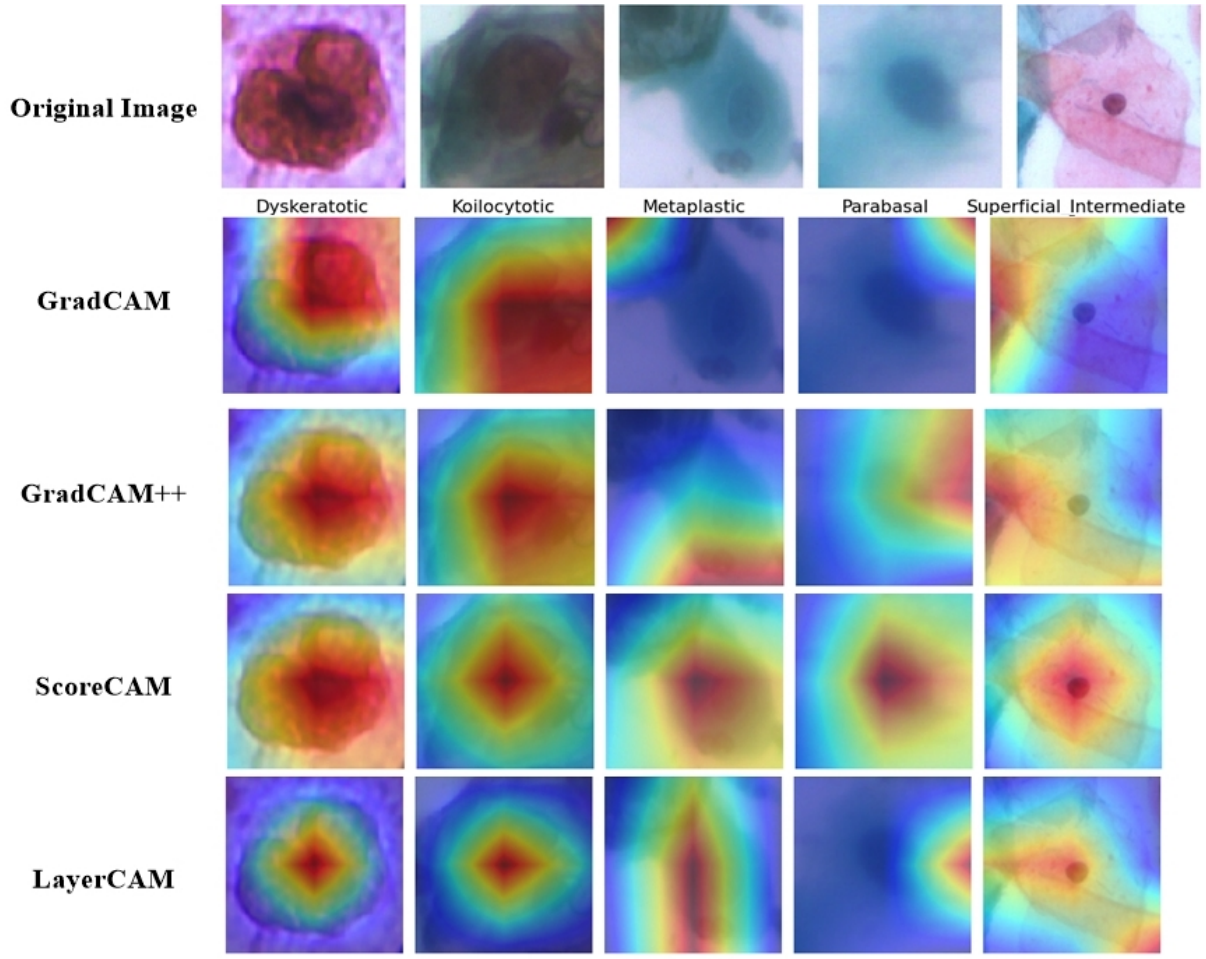


Figure 6.5: Five sample outputs of correctly classified instances of DenseNet169 using gradient-based XAI techniques

[2] https://github.com/keisen/tf-keras-vis

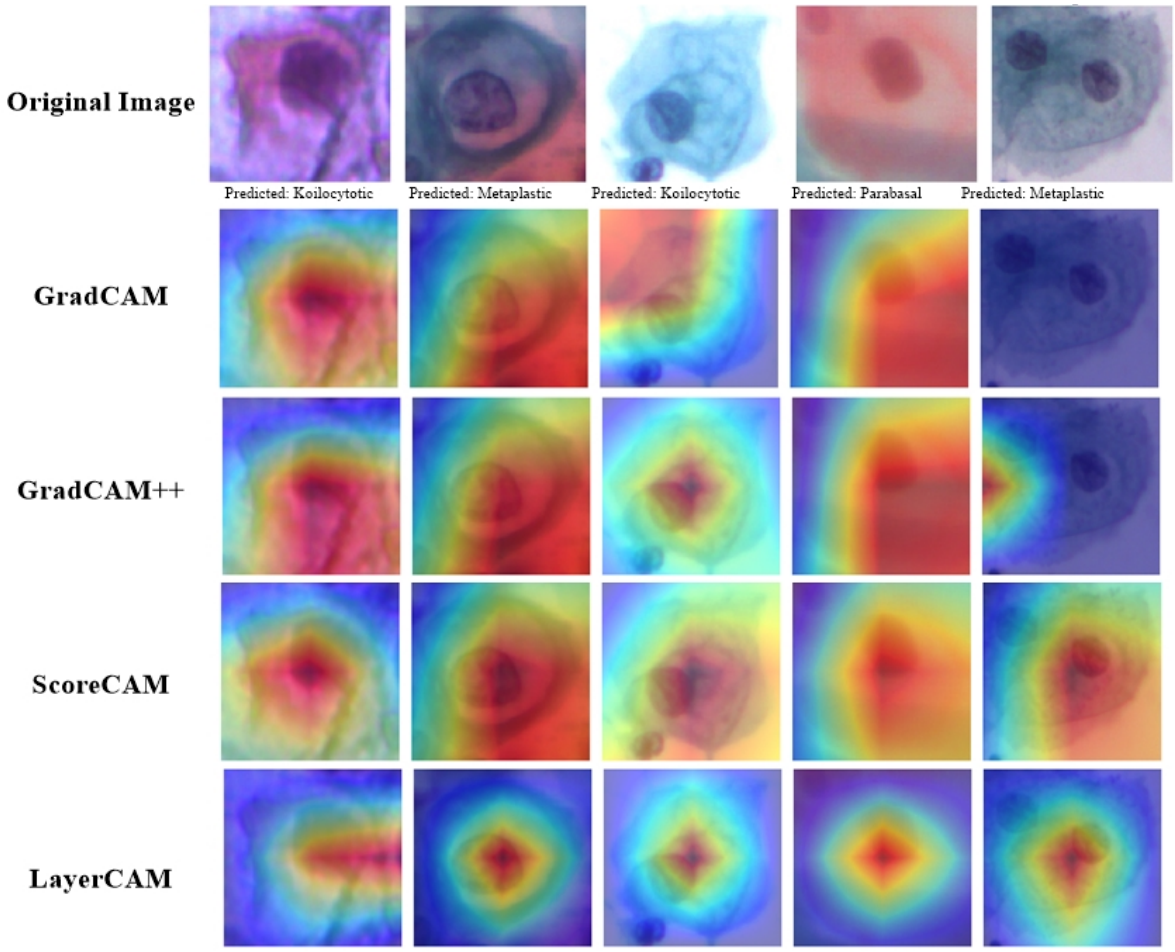


Figure 6.6: Five sample outputs of misclassified instances of DenseNet169 using gradient-based XAI techniques

Figure 6.5 represents five correctly classified images from five different classes and 6.6 represents five misclassified images where advanced gradient-based visualization techniques were employed, including GradCAM, GradCAM++, ScoreCAM, and LayerCAM. By applying these methods, crucial areas within the images were highlighted, shedding light on the regions that influenced the classification. These visualization techniques offer valuable insights into the inner workings of the model and the factors contributing to its decision-making process. In 6.5, GradCAM and GradCAM++ were unable to generate satisfactory heatmaps for the metaplastic and parabasal classes. Similarly, LayerCAM also struggled to produce an accurate heatmap for the parabasal class. However, ScoreCAM proved to be effective in generating a reliable heatmap for the parabasal class. The limitations observed in the performance of GradCAM, GradCAM++, and LayerCAM emphasize their difficulty in capturing the essential features and influential regions within the metaplastic and parabasal class images. Despite their shortcomings, these techniques still contribute valuable insights into the model's decision-making process for other classes. On the other hand, ScoreCAM emerged as a robust technique that successfully highlighted the crucial areas within the parabasal class images. Its ability to generate an accurate heatmap for this par-

ticular class showcases its effectiveness in identifying the significant features influencing the model's classification. The differential performance of these visualization techniques underscores the importance of employing multiple methods when analyzing and interpreting the decision-making process of a model. By using a combination of techniques like GradCAM, GradCAM++, LayerCAM, and ScoreCAM, we were able gain a comprehensive understanding of the model's behavior across different classes, identifying both strengths and weaknesses in its classification capabilities.

6.4.2 Perturbation-based Visualization

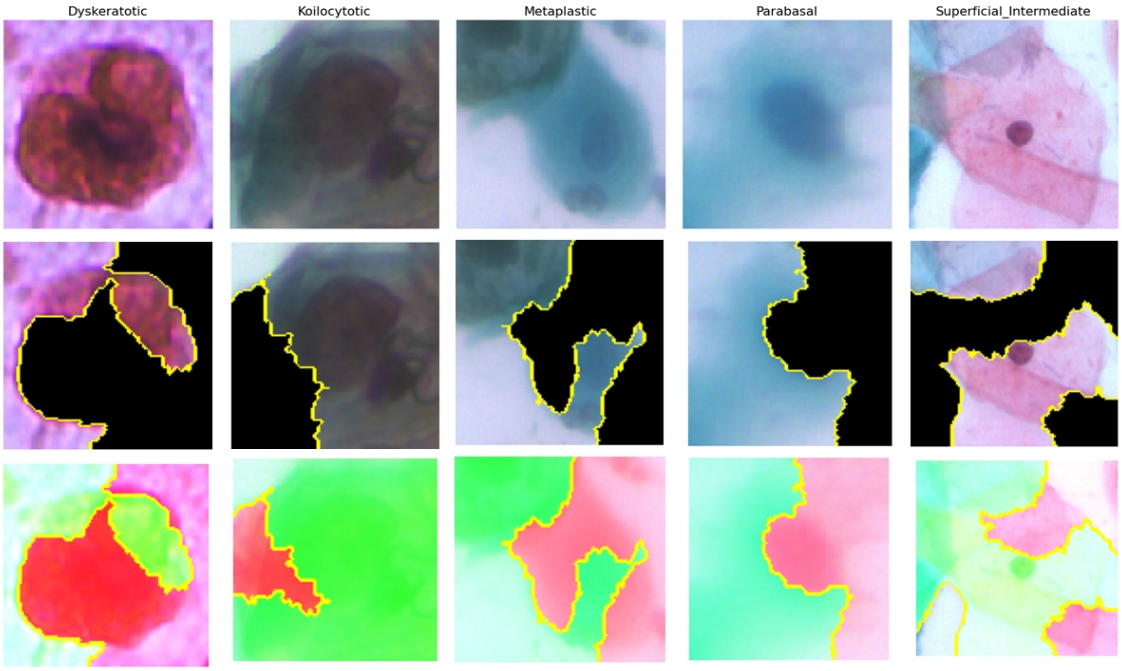


Figure 6.7: Five sample outputs of correctly classified instances of DenseNet169 using Perturbation-based Visualization technique LIME

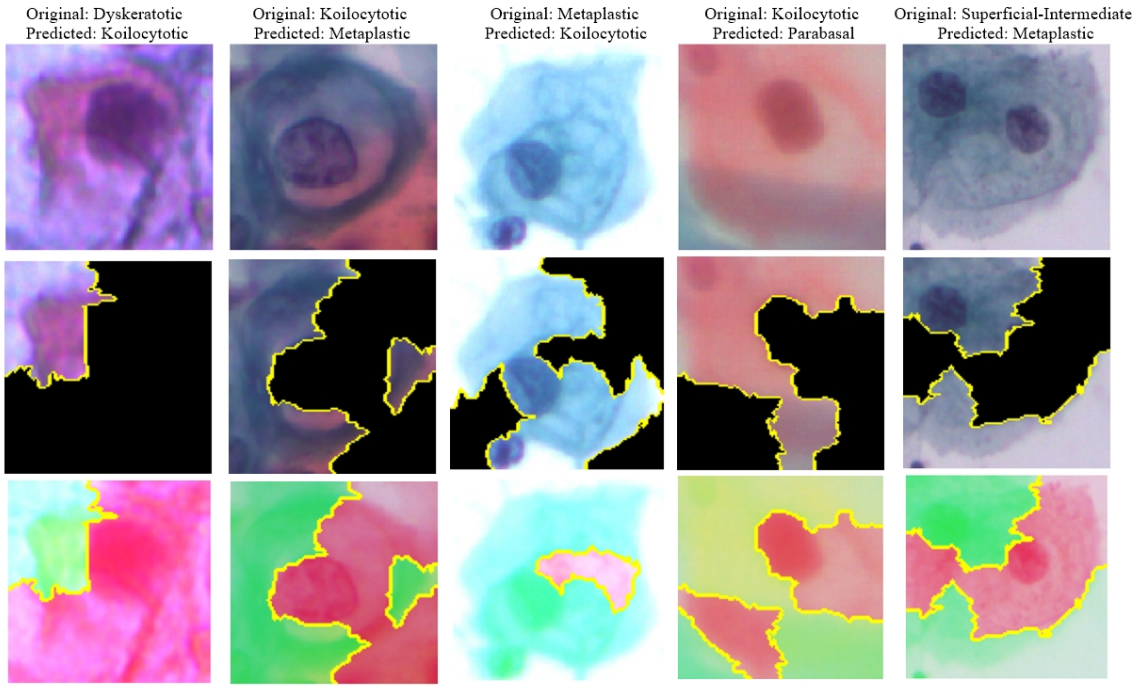


Figure 6.8: Five sample outputs of misclassified instances of DenseNet169 using Perturbation-based Visualization technique LIME

Figure 6.7 displays a set of five correctly classified images representing 5 different classes, while Figure 6.8 presents five misclassified images where the perturbation-based visualization technique, Local Interpretable Model-Agnostic Explanations (LIME) was used. LIME helped identify and highlight the important regions and features within these images that contributed to the incorrect classifications. By generating random perturbations of the input data, LIME identified the top superpixels that positively influence the predicted class. LIME generated two types of visualizations of the images: one that focuses solely on positive contributions and another that incorporates both positive and negative contributions. These visualizations allowed us to gain insights into potential weaknesses or biases present in the model's decision-making process. By highlighting the influential regions while preserving the rest of the image, LIME provides interpretable insights into the factors influencing the model's output. Moreover, LIME employed color-coded visualizations, with green representing the "pros" (positive contributions) and red representing the "cons" (negative contributions). This color scheme facilitates a clear understanding of the model's decision by visualizing the positive and negative aspects of the prediction. LIME's approach also gives interpretable insights into the model's decision-making process by showcasing the pixels that influenced the misclassifications.

Chapter 7

Discussion

7.1 Comparison with Previous Works

Rahaman et al. [11] introduced a hybrid deep feature fusion (HDFF) using VGG-16, VGG-19, ResNet-50, and XceptionNet models achieving an accuracy of 99.14%.Another work [5] achieved 96.96% by using a fuzzy distance-based ensemble approach. Manna et al. [12] presented ensemble methods achieving an accuracy of 95.43%. Tripathi et al. [4] proposed deep learning techniques to classify cells based on different stages of cancer cell growth achieving an accuracy of 94.89%. On the other hand, Among the five models used in our study, we noticed that different techniques produced high classification accuracy. By fine-tuning technique, The best accuracy of 97.17% was attained by the densenet169 model, followed by cost-sensitive learning with the same model at 97.29%. Additionally, using supervised contrastive learning highest accuracy of 97.29% learning with ResNet-50 and VGG-16 models was attained among four models. Table 7.1 shows a summary of the previous study and our study.

Table 7.1: Comparison with previous works

Author/Year	Methods	Accuracy
Rahaman et al./2021 [11]	**Hybrid deep feature fusion (HDFF) technique using VGG-16, VGG-19, ResNet-50, and XceptionNet**	**99.14%**
Pramanik et al./2022 [5]	Fuzzy distance-based ensemble approach using Inception V3, Inception ResNet V2, and MobileNet V2	96.96%
Manna et al./2021 [12]	Ensemble technique using Inception v3, DenseNet-169, and Xception	95.43%
Tripathi et al./2021 [4]	Fine-tuning (ResNet-152)	94.89%
Our study	Fine-tuning (DenseNet169)	97.17%
	Cost-Sensitive learning (DenseNet169)	**97.29%**
	Supervised Contrastive learning (VGG-16)	**97.29%**

From the comparison with previous studies, it is evident that our study has proven to perform better than many previous studies conducted on the same dataset.

7.2 Limitations and Future Works

In our study, we were unable to use many hyperparameters. Though to classify cervical cell images, multiple datasets are available; however, we were limited to working with only one dataset. In the future, we can explore various new aspects. For example, (1) Applying our methodology on different datasets to measure generalization performance, (2) Experimenting with different hyper-parameters to evaluate their impact on the classification performance of cervical cell images, (3) Finding a more accurate model for improving the performance of classification of cervical cell images, (4) Incorporating more Explainable AI techniques for better interpretability.

Chapter 8

Conclusion

Cervical cancer poses a significant global health challenge as the third-most common type of cancer and the leading cause of cancer-related deaths among women [1]. However, advancements in screening tests and automated evaluation systems offer promising avenues for prevention and early detection. In our study, we have focused on leveraging deep learning techniques for the classifying cervical cancer cells, aiming to contribute to the development of effective automated systems.

To achieve this goal, we have implemented 5 deep learning algorithms: VGG16, VGG19, ResNet50, MobileNetV2, and DenseNet169. Recognizing the importance of accurately identifying cervical cancer, we have incorporated cost sensitivity into our models. By making the models cost-sensitive, we have emphasized the correct classification of cancer cases, reducing the risks of both false positives and false negatives.

To evaluate how our models perform, we have used some evaluation metrics that are precision, recall, F1 score, and accuracy. These metrics provide comprehensive insights into the models' ability to classify cervical cancer cells accurately and reliably. Through the analysis of these metrics, we have assessed the strengths and limitations of each algorithm, enabling us to make informed decisions regarding their potential deployment in real-world scenarios.

Furthermore, we have utilized Gradient-based and Perturbation-based visualization approaches to enhance the interpretability of our automated cervical cancer detection system. These techniques have allowed us to gain a deeper understanding of the features and patterns that contribute to the identification of cancerous cells within cervical images. By visualizing the gradients and perturbations, we can establish greater trust in the decisions made by the automated system and provide clinicians with valuable insights for further analysis and diagnosis.

Overall, our study on the classification of cervical cancer with deep learning techniques highlights the potential of automated evaluation systems in the fight against cervical cancer. By

leveraging advanced algorithms and incorporating cost sensitivity, we have demonstrated promising results in accurately identifying cervical cancer cells. The utilization of visualization techniques further enhances the interpretability of our system, contributing to the overall trustworthiness and effectiveness of automated cervical cancer screening. Our findings provide a stepping stone towards the development of robust and reliable tools that can aid in the early detection and prevention of cervical cancer, ultimately saving numerous lives and reducing the burden of this public health concern.

References

[1] M. Arbyn, X. Castellsagué, S. de Sanjosé, L. Bruni, M. Saraiya, F. Bray, and J. Ferlay, "Worldwide burden of cervical cancer in 2008," *Annals of oncology*, vol. 22, no. 12, pp. 2675–2686, 2011.

[2] K. Duraisamy, K. Jaganathan, and J. C. Bose, "Methods of detecting cervical cancer," *Advances in Biological Research*, vol. 5, no. 4, pp. 226–232, 2011.

[3] Z. Lai and H. Deng, "Medical image classification based on deep features extracted by deep model and statistic feature fusion with multilayer perceptron," *Computational intelligence and neuroscience*, vol. 2018, 2018.

[4] A. Tripathi, A. Arora, and A. Bhan, "Classification of cervical cancer using deep learning algorithm," in *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*, pp. 1210–1218, IEEE, 2021.

[5] R. Pramanik, M. Biswas, S. Sen, L. A. de Souza Júnior, J. P. Papa, and R. Sarkar, "A fuzzy distance-based ensemble of deep models for cervical cancer detection," *Computer Methods and Programs in Biomedicine*, vol. 219, p. 106776, 2022.

[6] H. Panwar, P. Gupta, M. K. Siddiqui, R. Morales-Menendez, P. Bhardwaj, and V. Singh, "A deep learning and grad-cam based color visualization approach for fast detection of covid-19 cases using chest x-ray and ct-scan images," *Chaos, Solitons & Fractals*, vol. 140, p. 110190, 2020.

[7] M. Esmaeili, R. Vettukattil, H. Banitalebi, N. R. Krogh, and J. T. Geitung, "Explainable artificial intelligence for human-machine interaction in brain tumor localization," *Journal of Personalized Medicine*, vol. 11, no. 11, p. 1213, 2021.

[8] M. E. Plissiti, P. Dimitrakopoulos, G. Sfikas, C. Nikou, O. Krikoni, and A. Charchanti, "Sipakmed: A new dataset for feature and image based classification of normal and pathological cervical cells in pap smear images," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, pp. 3144–3148, IEEE, 2018.

[9] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," *Advances in neural information processing systems*, vol. 33, pp. 18661–18673, 2020.

[10] A. Ghoneim, G. Muhammad, and M. S. Hossain, "Cervical cancer classification using convolutional neural networks and extreme learning machines," *Future Generation Computer Systems*, vol. 102, pp. 643–649, 2020.

[11] M. M. Rahaman, C. Li, Y. Yao, F. Kulwa, X. Wu, X. Li, and Q. Wang, "Deepcervix: A deep learning-based framework for the classification of cervical cells using hybrid deep feature fusion techniques," *Computers in Biology and Medicine*, vol. 136, p. 104649, 2021.

[12] A. Manna, R. Kundu, D. Kaplun, A. Sinitca, and R. Sarkar, "A fuzzy rank-based ensemble of cnn models for classification of cervical cytology," *Scientific Reports*, vol. 11, no. 1, p. 14538, 2021.

[13] T.-C. Hsieh, C.-W. Liao, Y.-C. Lai, K.-M. Law, P.-K. Chan, and C.-H. Kao, "Detection of bone metastases on bone scans through image classification with contrastive learning," *Journal of Personalized Medicine*, vol. 11, no. 12, p. 1248, 2021.

[14] V. Ravi, "Attention cost-sensitive deep learning-based approach for skin cancer detection and classification," *Cancers*, vol. 14, no. 23, p. 5872, 2022.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[16] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

[17] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[19] S. H. Khan, M. Hayat, M. Bennamoun, F. A. Sohel, and R. Togneri, "Cost-sensitive learning of deep feature representations from imbalanced data," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 8, pp. 3573–3587, 2017.

[20] V. Ravi, H. Narasimhan, and T. D. Pham, "A cost-sensitive deep learning-based meta-classifier for pediatric pneumonia classification using chest x-rays," *Expert Systems*, vol. 39, no. 7, p. e12966, 2022.

[21] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification.," *Journal of machine learning research*, vol. 10, no. 2, 2009.

[22] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," *Advances in neural information processing systems*, vol. 29, 2016.

[23] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

[24] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks," in *2018 IEEE winter conference on applications of computer vision (WACV)*, pp. 839–847, IEEE, 2018.

[25] H. Wang, Z. Wang, M. Du, F. Yang, Z. Zhang, S. Ding, P. Mardziel, and X. Hu, "Score-cam: Score-weighted visual explanations for convolutional neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 24–25, 2020.

[26] P.-T. Jiang, C.-B. Zhang, Q. Hou, M.-M. Cheng, and Y. Wei, "Layercam: Exploring hierarchical class activation maps for localization," *IEEE Transactions on Image Processing*, vol. 30, pp. 5875–5888, 2021.

[27] M. T. Ribeiro, S. Singh, and C. Guestrin, ""' why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144, 2016.

[28] M. Rashida and M. A. Habib, "Quantitative eeg features and machine learning classifiers for eye-blink artifact detection: A comparative study," *Neuroscience Informatics*, vol. 3, no. 1, p. 100115, 2023.

[29] F. Jerbi, N. Aboudi, and N. Khlifa, "Automatic classification of ultrasound thyroids images using vision transformers and generative adversarial networks," *Scientific African*, p. e01679, 2023.

[30] M. Radmanesh, A. A. Rezaei, M. Jalili, A. Hashemi, and M. M. Goudarzi, "Online spike sorting via deep contractive autoencoder," *Neural Networks*, vol. 155, pp. 39–49, 2022.

[31] J. Li, D. Zhang, B. Meng, Y. Li, and L. Luo, "Fimf score-cam: Fast score-cam based on local multi-feature integration for visual interpretation of cnns," *IET Image Processing*, vol. 17, no. 3, pp. 761–772, 2023.

Appendix A

Dataset and Codes

Our dataset and the implementation of all experiments are available at- https://github.com/isha-67/CervicalCancerStudy.